Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 2406968, 9 pages
https://doi.org/10.1155/2018/2406968

Research Article

Study on Formal Modeling and Safety Verification of Train-to-Train Communication

Haonan Feng [1,2]

[1] *Signal and Communication Research Institute, China Academy of Railway Sciences, Beijing 10081, China*
[2] *The Center of National Railway Intelligent Transportation System Engineering and Technology, Beijing 10081, China*

Correspondence should be addressed to Haonan Feng; fhn02212005@163.com

Received 6 December 2017; Revised 20 March 2018; Accepted 23 April 2018; Published 4 June 2018

Academic Editor: Li Zhu

VBTC (vehicle-to-vehicle communication based train control) has gradually become an important research trend in the field of rail transit. This has resulted in advantages of decreasing the number of pieces of wayside equipment and improving the efficiency of real-time system communication. Characteristics and mechanism of train-to-train communication, as key implementation technology of safety critical system, are given and discussed. A new method, based on the LTS (labelled transition system) model checking, is proposed for verifying the safety properties in the communication procedure. The LTS method is adapted to model system behaviours; analysis and safety verification are checked by means of LTSA (labelled transition system analyzer) software. The results show that it is an efficient method to verify safety properties, as well as to assist the complex system's design and development.

1. Introduction

With rapid development of urbanization, demands from increased living standards and travel lead to requirement of rapid development of urban transit transport. Signal system, as an important role of traffic safety, is to ensure traffic safety, shorten the operation interval, and improve traffic efficiency [1]. CBTC (communication based on train control) system adding wireless communication technology into traditional ground interlocking control system, which makes moving block operation mode under double protection of ground ATP (automatic train protection) and vehicle ATP, provides protective speed based on real-time speed from vehicle [2, 3]. CBTC system has a great advantage in terms of transportation capacity and system safety, and its great development in urban transit signalling system contributes to lower the urban traffic pressure [4, 5].

The next generation of train control system welcomes a new research field that follows on from the mature development of CBTC system and modern communication technology. VBTC system is a good example in attractive research trend [6]. The system is designed to improve train automotive control, reduce system coupling for better efficiency, and lower cost [7]. Train-to-train communication is the main supporting architecture in VBTC, whose simple structure has the advantage of decreasing equipment number and lowering construction and maintenance cost. Direct communication among trains in real time should decrease the impact from delay caused by undirected wireless communication and improve overall system efficiency[8, 9].

Although VBTC brings out innovative results by optimizing communication methods, it originally belongs to safety critical system, which requires design phase needed to be verified carefully to the extent of safety and security. Formal modeling is the process of converting the natural description language into a formal description language during the system's requirement design phase, which can improve the completeness and consistency of the system design. According to the description of the system characteristics, it can be divided into two categories: one is used to describe the system sequence characteristics based on mathematical analysis methods, such as Z method [10], B method [11], and VDM method [12]; the other method is preferable to describe concurrent characteristics of system, such as Petri net [13–15], finite automata [16, 17], and LTS [18].

Different from other formal methods, LTS has the unique ability to analyze the countably infinite set of states and actions of system and infinite interactive branching, so it is

appropriate for reactive and concurrent system behaviour. It provides method to trace state transit behaviour, identify observable states, and evaluate sequence of performed actions. As an ideal tool to verify concurrent and complex system, it gains a great achievement in the application of verifying safety protocol RSSP-1 [19] and needs of ERTMS/ETCS system [20]. This paper focuses on formal modeling and verification based on LTS method for train-to-train communication logic in the architecture of VBTC.

In the rest of this paper, we firstly define and describe new architecture of VBTC and make comparison with CBTC in Section 2. In Section 3, LTS and its auxiliary analysis tool LTSA software are introduced and adopted to model the behaviours of train-to-train communication. Safety property of train-to-train communication is verified in Section 4. Comparisons in the aspect of application and performance are listed using LTS method and traditional human checking method in Section 5. Finally, the conclusions are presented in Section 6.

2. The Architecture of VBTC

VBTC system is shown in Figure 1. Its five cardinal parts are central ITS (intelligent train supervision system), TMC (train management center), OC (object controller) belonging to wayside systems, IVOBC (intelligent vehicle on board controller) installed in vehicle, and DCS (data communication system), which provides data transmission link between static equipment and moving train. Main functions of each system are illustrated as follows.

Central ITS is in charge of train operation schedule, monitoring all the conditions of other systems, and processing safety related events. All the trains on line should interact with central ITS based on periodic and reliable communication. Central ITS checks status of every section to make sure safety of the whole line operation. It will provide all the necessary information required for trains to identify what trains are in front or behind.

TMC is responsible for the storage and distribution of electronic map, system configuration data, and temporary speed limits. Its communication mode should be aperiodic or passive response.

OC is a new kind of wayside equipment control unit. It collects and sends status of control objects in response to request of trains and central ITS; at the same time, it receives control information as operation input. Passive communication is interactive mode of this type of unit.

IVOBC is the core of VBTC system. It inherits all the features of ATP (automatic train protection) and ATO (automatic train operation) and V2I (vehicle-to-infrastructure) communication in CBTC system; IVOBC also develops the new functions of route plan, route safety protection, and moving authority calculation.

DCS should support multiple communication protocols as an important bridge link for IVOBC and other systems, for communication standards should vary adaptively with distance parameters in the adjacent trains. For instance, a train should communicate with other trains by the communication form of LTE-V2X in long range, and it chooses direct-to-direct communication form when range is less than 1km.

The property differences between VBTC and CBTC system are compared in the five aspects of system components, train-to-train communication mode, train control mode, wayside equipment, and daily maintenance (see Table 1).

3. Formal Modeling and Verification Methodology Based on LTS

The communication cooperation among trains has the characteristics of complex concurrency and logic synchronization. Therefore, LTS method is selected by the advantage of accurate description of system behaviour in real time and concurrency. The train-to-train communication cooperation strategy in VBTC is modeled by LTS method, which provides a basis for subsequent simulation and verification analysis.

3.1. Formal Definition of LTS. Let S represent the universal set of states including a designated error state π. L is the universal set of labels and acts as $D = L \cup \{\tau\}$, where τ denotes an internal actions that cannot be observed by the environment of an LTS.

An LTS of a process P is a quadruple$< Q, A, T, q_0 >$, where

(1) Q is a countable finite and nonempty set of states,

(2) A is a countable set of labels $A = \alpha P \cup \{\tau\}$, where αP denotes the alphabet of P, and $\alpha P \subseteq L$,

(3) $T \subseteq Q - \{\pi\} \times A \times Q$ denotes a transition relation that maps from a state and an action onto another state,

(4) $q_0 \in Q$ indicates the initial state of P.

The only LTS that is allowed to have the error state π as its initial state is $< \{\pi\}, D, \{\}, \pi >$, named E. The alphabet of this process $\alpha E = L$.

An LTS $P =< Q, A, T, q_0 >$transits with action $a \in A$ into P', denoted as $P \xrightarrow{a} P'$, if

(1) $P' =< Q, A, T, q_0' >$,where $q_0' \neq \pi$ and $(q, a, q_0') \in T$,

(2) $P' = E$, and $(q, a, \pi) \in T$.

$P \xrightarrow{a}$ means that $\exists P'$such that $P \xrightarrow{a} P'$.

End states $Z \subseteq S$ such that an LTS $P =< Q, A, T, q_0 >$ is terminating if there is a state $e \in Z$ and $!\exists(e, a, q) \in T$ for all $a \in A$.

LTSA software, as an auxiliary analysis tool based on LTS, can animate and check the behaviour of the overall system before it is implemented. It focuses on the aspect of concurrency, provides methods to model animation to visualize system behaviour, and gives mechanical verification of system properties including safety and progress. LTSA program supports process algebra notation, such as FSP (finite state process), for concise description of system component behaviour [21].

3.2. Train-to-Train Communication Mechanism. Train-to-train communication can be broken down into a set of

Table 1: Comparison between VBTC and CBTC system.

Property	VBTC	CBTC
System components	Central ITS TMC IVOBC DCS OC	Central ATS (automatic train supervision) DSU (digital storage unit) VOBC (vehicle on board controller) DCS ZC (zone controller) CI (computer interlocking)
Train to train communication mode	Direct radio communication between trains	Indirect communication between trains, interactive information needs to be relayed by wayside equipment
Train control mode	IVOBC is center of system, ground only provides auxiliary functions, such as schedule plan, emergency procedure	Train operation is controlled by ATP from ground and vehicle
Wayside equipment	Little wayside equipment, only balises, switch controller	Amount of axle counting, balises, semaphores
Daily maintenance	less maintenance data mainly involving of train	Amount of maintenance data including train, wayside equipment

simpler activities, its mechanism should include at least three basic functions: (a) identification of train in front, (b) communication status management between train and the train in front, and (c) communication status management between train and the following train. These activities are not executed one after the other in a strictly fixed order in the physical world. In fact, it is noticing that these activities are permitted to overlap or occur concurrently in certain scenarios.

(a) Identification of Train in Front. It is accessible to get numbers ID information of other trains when a train enters recognizable region in touch with central ITS. After a train confirms all the trains in the same region in the mode of calling and answering, it begins to identify and confirm the train in front. The procedure of this stage is shown in Figure 2.

(b) Communication Status Management between Train and the Train in Front. Implementation of the stage should be divided into four steps as follows:

(1) Get the train ID in front and check tracing condition requirements.

(2) Send the tracing establishment request message when tracing condition is satisfied.

(3) Wait and receive tracing establishment reply from train in front; the communication between train and the front should be closed if communication time is over the designed parameter of time gate.

(4) It is successful to enter tracing mode after train accepts reply from the train in front; otherwise the train should resend tracing establishment request message in the case that communication between train and the front status is still valid, or the train rebuild tracing link from the beginning if communication time is over the designed parameter time gate.

The procedure of this stage is shown in Figure 3.

(c) Communication Status Management between Train and the Following Train. It is similar to the previous stage. The tracing mode is triggered by the following train; confirmation message is sent by the train in response to the tracing request of the following train. Train will check condition after receiving message. The procedure of this stage is shown in Figure 4.

3.3. Simulation and Verification. Designed to be easily machine readable, FSP is a simple algebraic notation to describe process models; each description of state in LTS has a corresponding FSP description. The semantics of basic FSP can be easily defined in terms of LTS, and LTSA software has the function of depicting the LTS by FSP language as a graph. In the following, their correspondence is defined by the function in

$$lts : E \longrightarrow \xi \tag{1}$$

where E is the set of FSP process expressions and ξ represents the set of LTSs. The function lts is defined inductively on the structure of FSP process expressions.

After analyzing train-to-train communication workflow in the Section 3.2, action sets of system behaviour are illustrated by alphabet sets A, while system process S and intermediate transition process variables M_i,(i=1,2,...9) should be defined in FSP language as follows.

A={*inichk, inisysOK, inisysBad, sndReqTrain, rcvFdReqTrain, checkTrainID, rcvReqTrain, reqTraintimeout, trainAlarm, sndITS, isTrainPos, trainisFront, trainisBack, sndBldTracReq, bldTracReqTimeout, rcvBldTracAns, checkTracCond, tracCondisOK, tracCondNotOK, setTracMode, selfFixBlock, selfTracingMode, sndDismissTracReq, dismissTracReqtimeout* }

$$S = M_1,$$
$$M_1 = (\ inichk\text{->}inisysOK\text{->}M_2 \mid inichk\text{->}inisysBad\text{->}M_3\ ),$$

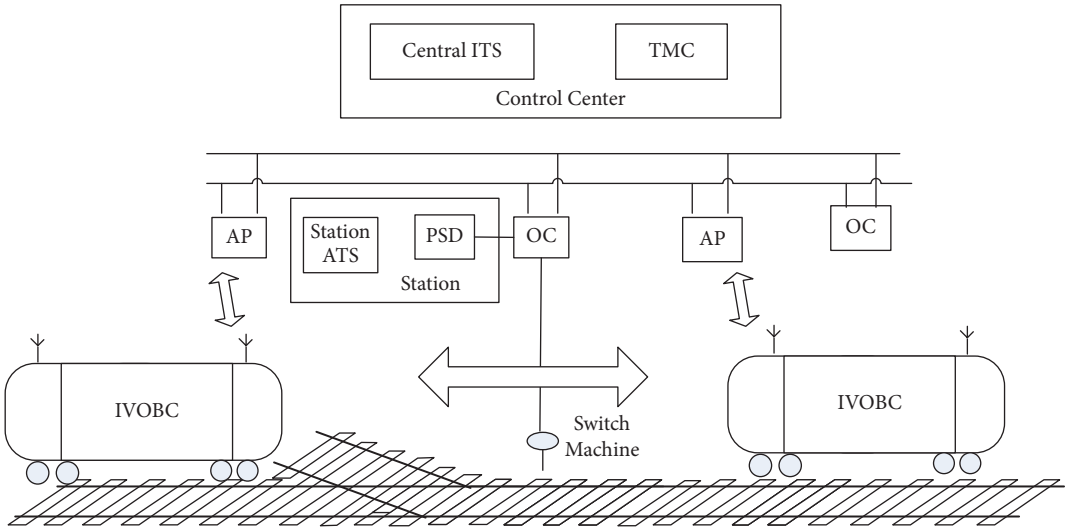

Figure 1: Architecture of train-to-train based train control system.

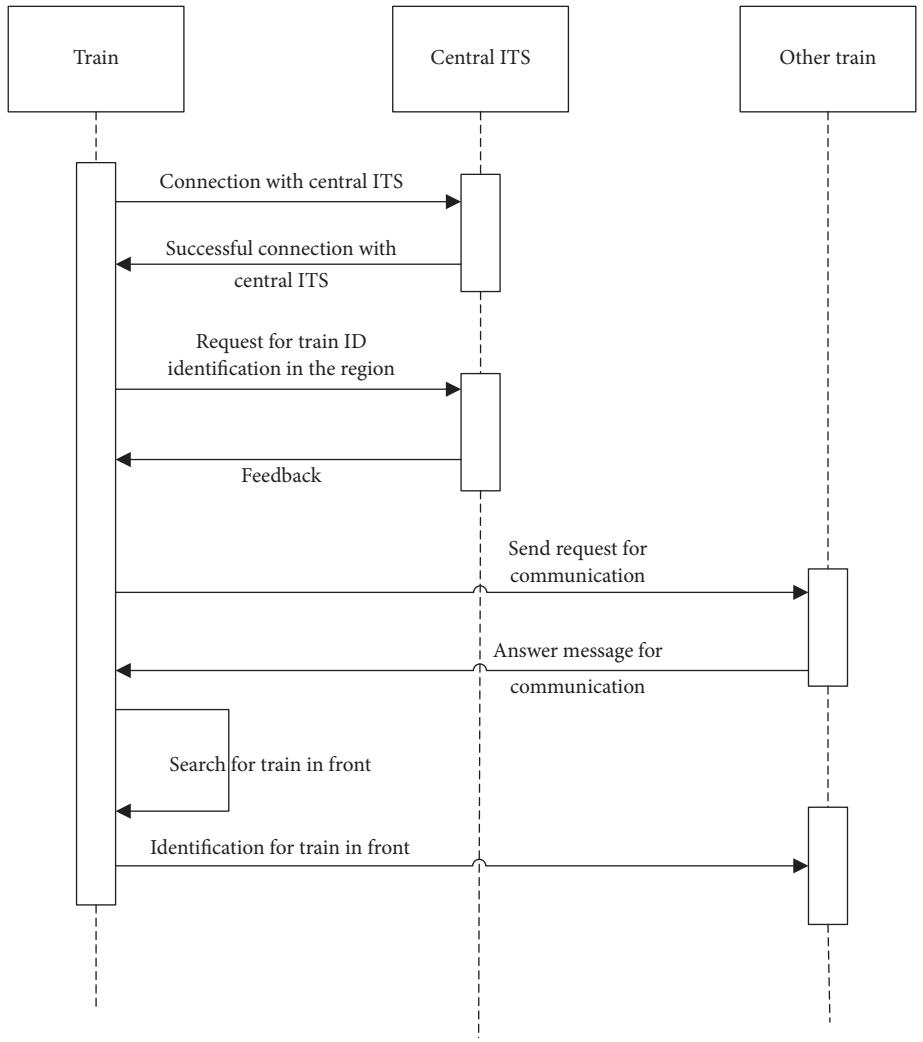

Figure 2: Time sequence of train in front identification.

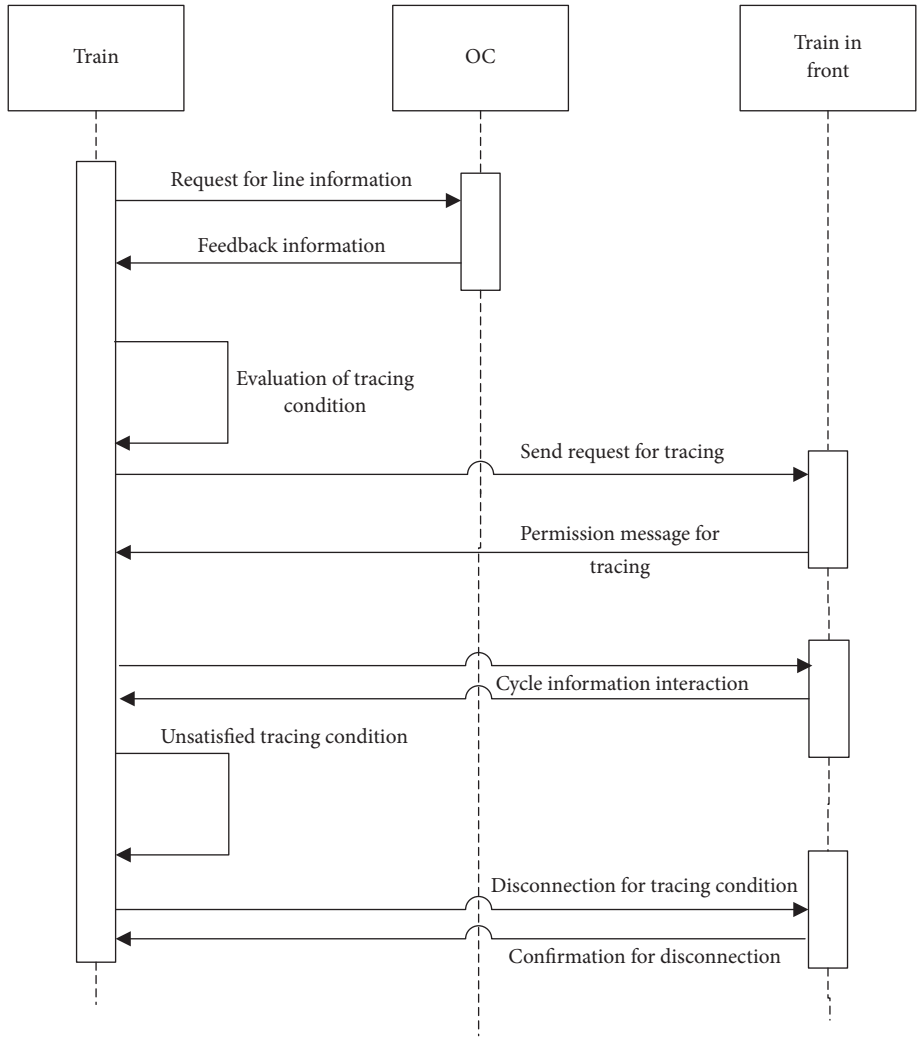

Figure 3: Holding process of tracing mode.

M_2 = (*sndReqTrain->rcvFdReqTrain->checkTrainID->*M_4 | *rcvReqTrain-> checkTrainID->*M_4 | *sndReqTrain ->reqTraintimeout->*M_3),

M_3= (*trainAlarm->sndITS->*M_1),

M_4= (*isTrainPos->trainisFront->*M_5 | *isTrainPos->trainisBack->*M_6),

M_5 = (*sndBldTracReq->* M_6 | *sndBldTracReq ->bldTracReqTimeout ->* M_3),

M_6 = (*rcvBldTracAns->*M_8),

M_7 = (*checkTracCond->tracCondisOK->*M_8 |*checkTracCond->tracCondNotOK->* M_9),

M_8 = (*setTracMode->selfFixBlock->*M_2 | *setTracMode->selfTracingMode->*M_5),

M_9 = (*sndDismissTracReq->rcvDismissTracAns ->*M_8 | *sndDismissTracReq->dismissTracReqtimeout ->*M_3).

Detailed explanations of actions in FSP language are shown in Table 2.

By means of LTSA software, the graphic transition actions and processes of train-to-train communication model are shown in Figure 5. The nodes in the Figure 5 represent states of the system, and transitions are depicted by connecting edges. It is noticed that no error label -1 is shown in these model checking result. The verification results by LTSA in Figure 6 give that there is no deadlock or logic violations in 19 states before the model is implemented into executable codes.

4. Safety Verification

In the design phase of VBTC, there exist some dangerous situations that should never not be allowed to occur. Designers and programmers should take careful and thoughtful actions to these dangerous situations. Fault-oriented safety principle is convention in the engineering field of rail transit. The prohibited system requirements should be listed and fulfilled in the final implementation of system.

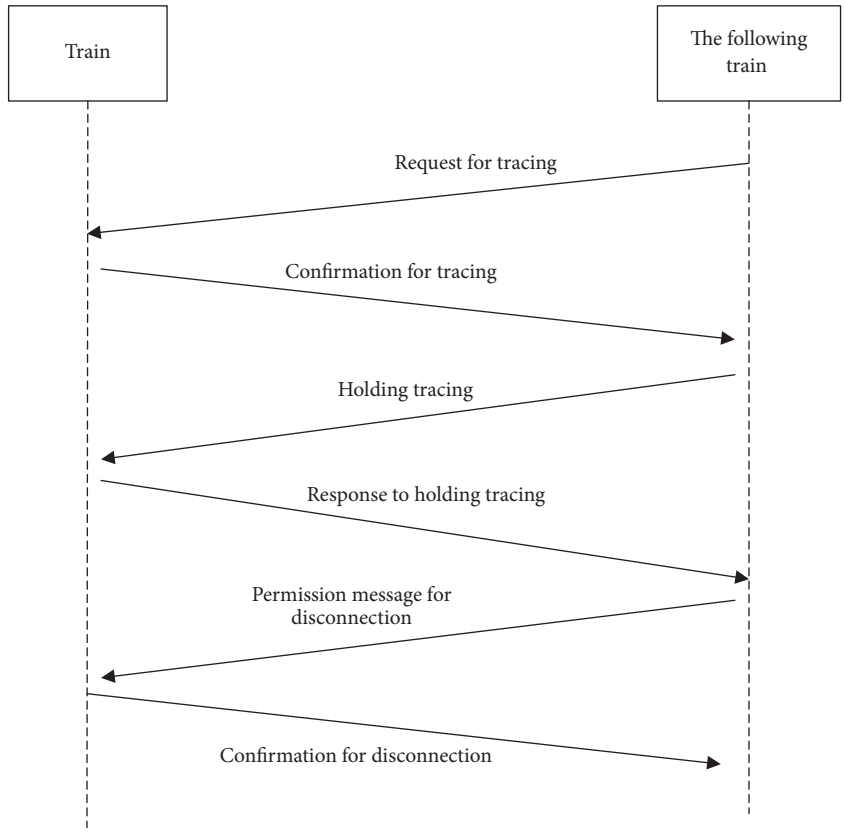

Figure 4: Holding process of tracing mode between train and following train.

The following actions are prohibited; the safety aspects should be verified strictly as safety property in the system model.

(1) Send request message to other train when the initialization check of self system fails.

(2) Send request message to other train when train alarms for error.

(3) Send build tracing request message to other train in the case of timeout for request message.

(4) Tracing condition should not be satisfied in the case of timeout for built tracing request.

(5) The mode is not allowed to be set tracing mode in the case of timeout for tracing request message.

(6) The mode is not allowed to be set tracing mode when tracing condition is unsatisfied.

The safety property in these unacceptable behaviours above modeled by FSP language is follows:

property TrainSafety = *Safe*,
Safe =(*inisysBad* -> *sndReqTrain* -> *Safe*
|*trainAlarm*-> sndReqTrain -> *Safe*
|*reqTraintimeout* -> *sndBldTrackReq* -> *Safe*
|*bldTrackReqTimeout* -> *trackCondisOK*-> *Safe*
|*dismissTrackReqtimeout* -> *selfTracingMode* -> *Safe*
|*trackCondNotOK* -> *selfTracingMode*-> *Safe*).

It appears a mistake state labelled '–1' in the system in Figure 7, which means that, in all transition traces begun from initial state, there are no exceptional transition lists from (1) to (6). It can be confirmed that the model of system satisfies the safety requirements.

5. Application and Performance

The formal modeling and verification method presented above is used in the deployment of train-to-train communication program. In order to evaluate the efficiency of the LTS modeling method, two independent developing teams with the same labour and similar technical experience are assigned to fulfill the program.

Both teams start the task simultaneously, and eight hours is their work time in one day. The entire developed procedure is divided into two stages, developing phase and testing phase. The main task of the first phase is to convert executable code from model based on the system requirement; testers in the second phase focus on errors occurring in the executable code, analysis, and classification of their reason. Operating time and categorized bugs are recorded in summary as important evaluation parameters in Table 3.

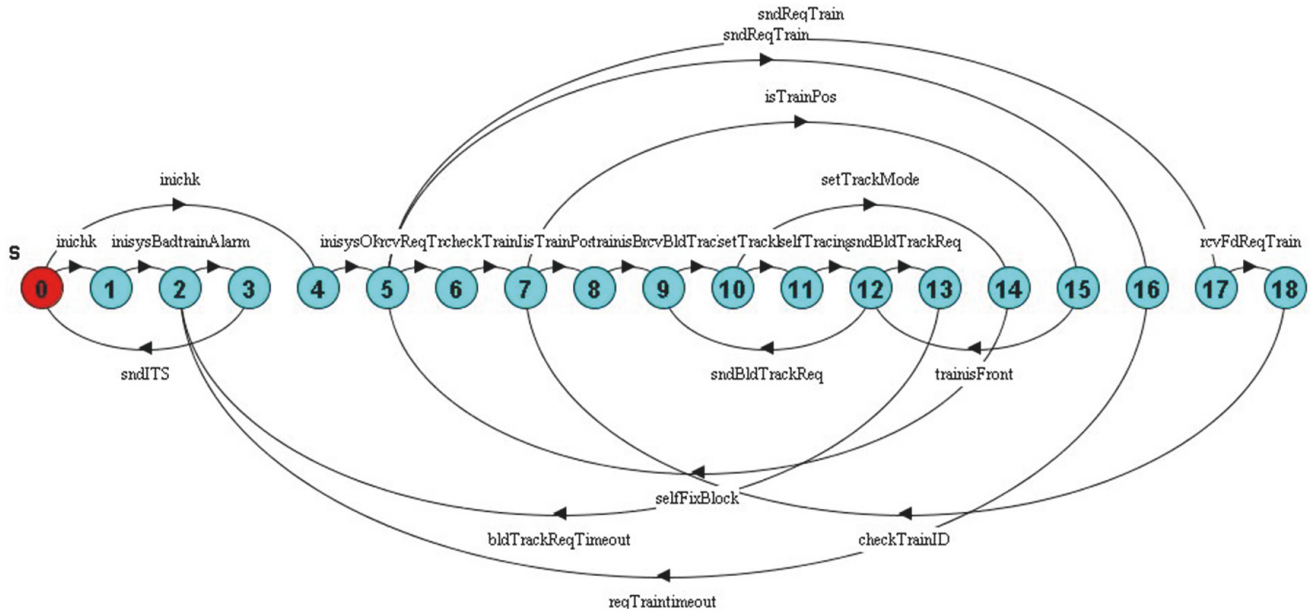

Figure 5: Train-to-train communication model.

Table 2: Actions description in system mode transition model.

Action	Description	Action	Description
inichk	Self system is in initialization check	*sndBldTracReq*	Send build tracing request message
inisysOK	The initialization check of self system is OK	*bldTracReqTimeout*	Timeout for built tracing request
inisysBad	The initialization check of self system fails	*rcvBldTracAns*	Receive build tracing answer message
sndReqTrain	Send request message to trains	*checkTracCond*	Check the condition of tracing mode
rcvFdReqTrain	Receive feedback message from trains	*tracCondisOK*	Tracing condition is still satisfied
checkTrainID	Check train ID	*tracCondNotOK*	Tracing condition is unsatisfied
rcvReqTrain	Send request message to trains	*setTracMode*	Evaluate the train mode
reqTraintimeout	Timeout for request message	*selfFixBlock*	The mode is set fixed blocked mode
trainAlarm	Train alarm for error	*selfTracingMode*	The mode is set tracing mode
sndITS	Send error message to central ITS	*sndDismissTracReq*	Send request message for dismissing tracing
isTrainPos	Evaluted train postion	*rcvDismissTracAns*	receive request message for dismissing tracing
trainisFront	Train is in front	*dismissTracReqtimeout*	Timeout for tracing request message
trainisBack	Train is behind		

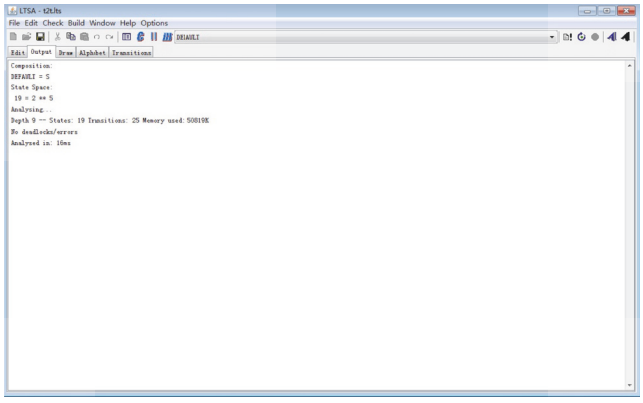

Figure 6: Safety verification report of train-to-train communication model.

It turns out to be more efficient and time-saving to use the LTS method compared with the conventional method. Approximately 51% of the total developing time is saved. Less time consumption did not lead to quality reduction. On the contrary, formal modeling and verification make further improvement on the quality. By analyzing errors in the test phase, we find that 17 out of 25 bugs by traditional method are derived from logic errors, and the remaining are due to data configuration. Statics show that logic bugs are attributed to two reasons in the developing phase. The first one is uncertain and ambiguous system requirements, which give programmers vague understanding of system behaviour; the other reason is some dangerous system behaviour unpredicted by programmer by the limits of manual screening. All these traps are avoided by LTS model developer. All the seven data bugs in LTS model come from wrong engineering data, none from control area of LTS model. Nearly 72% reduction in

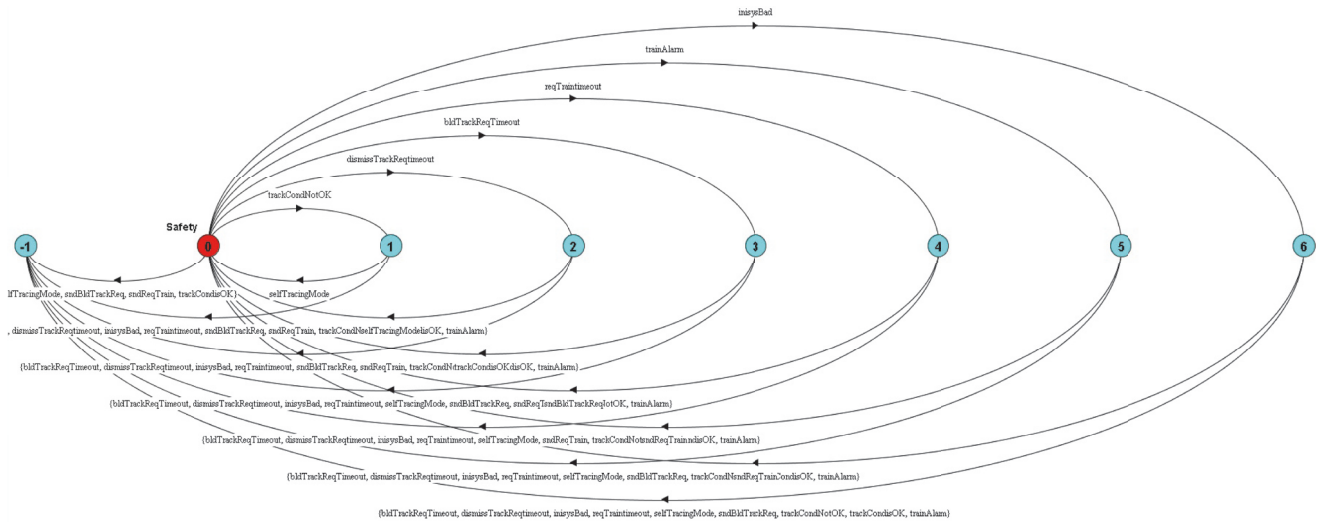

Figure 7: Safety property verification chart of train-to-train communication.

Table 3: Comparison of LTS method and traditional human checking.

Development method	Number of logic bugs	Number of data bugs	Total number of bugs	Number of developers	Developing time (hour)	Number of testers	Test time (hour)	Total Time (number of people×hour)
LTS Model	0	7	7	6	24	7	12	228
Traditional Method	17	8	25	6	48	15	12	468

the design flaws proves the powerful ability of LTS formal method. The comparison experimental results indicate that LTS model absolutely helps developer shorten design cycle and improve software quality.

6. Conclusion

With the final goal of allowing designers to describe train-to-train communication safely and efficiently, this paper presents a novel method for modeling and verifying such real-time systems. The train-to-train communication mechanism is described in FSP language which introduces logic properties in a simple and explicit manner. LTS model for train-to-train communication is built, and the function requirements as well as safety properties are verified with the aid of LTSA software. In the system simulation, LTSA verification software shows a powerful tool for complex systems by visualizing train-to-train LTS model. It thoroughly animates and checks whether the specification of candidate system satisfies the properties required, and it supports specification animation to interactively research on system behaviour. Simulation and verification results indicate that LTS method provides great assistance for designers to develop more efficient and reliable real-time systems.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research work has been supported by Key Project of China Academy of Railway Sciences (1651TH5302, 1651TH6603, and 1752TH1004); the National Natural Science Foundation of China (no. 61603026); Beijing Natural Science Foundation (L171004).

References

[1] B. Ning and C. Liu, "Technology and application of train operation control system for china rail transit system," *Journal of the China Railway Society*, vol. 39, no. 2, pp. 1–9, 2017.

[2] L. Zhu, F. R. Yu, and B. Ning, "An optimal handoff decision algorithm for Communication-Based Train Control (CBTC) systems," in *Proceedings of the 2010 IEEE Vehicular Technology Conference (VTC 2010-Fall)*, pp. 1–5, Ottawa, ON, Canada, September 2010.

[3] L. Zhu, F. R. Yu, B. Ning, and T. Tang, "Cross-layer handoff design in MIMO-enabled WLANs for Communication-Based Train Control (CBTC) systems," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 719–728, 2012.

[4] L. Zhu, F. R. Yu, B. Ning, and T. Tang, "Design and performance enhancements in communication-based train control systems with coordinated multipoint transmission and reception," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1258–1272, 2014.

[5] L. Zhu, F. R. Yu, B. Ning, and T. Tang, "Handoff management in communication-based train control networks using stream control transmission protocol and IEEE 802.11p WLANs,"

EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, pp. 1–16, 2012.

[6] J. K. Xu, "Analysis of novel CBTC system based on train to train communication," *Railway Signaling and Communication*, vol. 50, no. 6, pp. 78–80, 2014.

[7] C. Li, *Research on Train Communication Cooperation under V2V Communication Environment based on State Machine*, Beijing Jiaotong University Master Degree, 2017.

[8] H. Du, J. G. Sun, and Q. Zhang, "A new generation of CBTC system without CI and ZC," *Urban Rapid Rail Transit*, vol. 30, no. 4, pp. 91–95, 2017.

[9] D. Briginshaw, "Alstrom's simplified CBTC technology to debut in lille," *International Railway Journal*, vol. 53, no. 1, pp. 23-24, 2013.

[10] N. Zafar, "Formal dynamic operational model of RIS components," *International Journal of Computer Science and Network Security*, vol. 11, pp. 91–97, 2011.

[11] H. D. Tong, B. Ning, and H. F. Wang, "Research on event-B based modelling and verification of interlocking route control," *Railway Computer Application*, vol. 22, no. 6, pp. 57–61, 2013.

[12] N. A. Zafar, "Formal model for moving block railway interlocking system based on un-directed topology," in *Proceedings of the 2nd Annual International Conference on Emerging Techonologies 2006, ICET 2006*, pp. 217–223, Peshawar, Pakistan, November 2006.

[13] H. Song and E. Schnieder, "Validation, verification and evaluation of a train to train distance measurement system by means of colored petri nets," *Reliability Engineering & System Safety*, vol. 164, pp. 10–23, 2017.

[14] E. Schnieder, L. Schnieder, and J. Müller, "Conceptual foundation of dependable systems modelling," *2nd IFAC Workshop on Dependable Control of Discrete Systems*, vol. 42, no. 5, pp. 198–202, 2009.

[15] M. M. Zu Hörste and E. Schnieder, "Modelling and simulation of train control systems using petri nets," in *FMRail Workshop*, 1999.

[16] Y. Dong and X. J. Gao, "Method for generating formal interlocking software model based on scenario," *Computer Science*, vol. 42, no. 1, pp. 193–195, 2015.

[17] X. Y. Zhao, R. J. Cheng, and Y. Cheng, "Formal modeling and parameter analysis method for train control system based on hybrid unified modeling language," *Journal of the China Railway Society*, vol. 38, no. 11, pp. 80–87, 2016.

[18] X. Wang, Z. W. Xu, and M. Mei, "A software safety verification method based on model checking," *Journal of Wuhan University (Natural Science Edition)*, vol. 56, no. 2, pp. 156–160, 2010.

[19] M. Meng, X. Zhongwei, W. Xi, and W. Yongbing, "Model checking-based safety verification for railway signal safety protocol-I," *International Journal of Computer Applications in Technology*, vol. 46, no. 3, pp. 195–202, 2013.

[20] M. Ghazel, "Formalizing a subset of ERTMS/ETCS specifications for verification purposes," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 60–75, 2014.

[21] J. Magee and J. Kramer, *Concurrency State Models and Java Programs*, John Wiley Sons Ltd, Chichester, England, 1999.