Modeling and verification of CTCS-3 level code sequence verification scene based on UPPAAL

Wei Xiao1*, Wenzhen Kuang2
1School of Automation & Electrical Engineering, Lanzhou Jiaotong University, Lan Zhou, Gan Su, 730070, China
2Gansu Research Center of Automation Engineering Technology for Industry & Transportation, Lan Zhou, Gan Su, 730070, China
*Corresponding author’s e-mail: 1084547194@qq.com

Abstract. Taking the code sequence verification scenario of the autonomous CTCS-3 train control system as an example, on the basis of introducing the code sequence verification function, it is proposed to take the security function attributes, functionality and time requirements as the verification objectives, at last, the workflow of the scheme is analyzed in detail, and the behavior and action of each module of the system are standardized by the timed automata language. The timed automata network model of the workflow sequence of the code sequence verification scheme is established, which lays the foundation for the later verification. Then, the nature of the system is verified by the UPPAAL verification tool. The research shows that: in the process of model establishment, strictly in accordance with the specification content, the description of the specification is accurate, and at the same time, the modeling and verification method of the train control system based on the time automaton can effectively study the security, timing and characteristics.

1. Introduction
China's CTCS-3 train operation control system is an extremely complex safety-critical system. The CTCS-3 train control system specifications are the basis for the design and development of the CTCS-3 train control system[1]. Norms that rely on experience and intuition alone inevitably have certain loopholes or security risks. In addition, there may be ambiguity in the specifications portrayed in natural language, which adversely affects the design and development of the system. Because of the above two reasons, it is particularly important to model and formalize the CTCS-3 train control system specifications [2]. Many scholars have done a lot of work on the modeling and formal verification of train control systems. However, the common problems of current modeling and verification methods are: (1) The starting point of most modeling and verification methods is a specific model of the train control system, but they cannot prove the relationship between the model and the specifications, even if they can Prove the correctness of the model, and cannot deduce the correctness of the specification; (2) The lack of scientific conversion rules between modeling and verification, or the conversion rules are incomplete, and the verification process is strongly subjective; (3) Establishing backtracking and tracking of verification results makes it very difficult to locate errors in verification. Based on the above reasons, this paper proposes a strict modelling and formal verification method for CTCS-3 train control system specifications[3].
2. Code sequence verification scenario model description

2.1. Scenario overview and security requirements
As shown in Figure 1, the train operates in the C3 level full monitoring mode. At this time, the front side of the station is open and passes through the access road, and the traffic authorization (MA) extends to the outbound station until the section[4]. When the vehicle-mounted device is close to the pit stop signal, and the received code sequence is L3 and below, the vehicle-mounted device uses the code sequence under C2 to perform the code sequence verification function, and compares the length of the driving permission and track circuit information permission sent by RBC, and shortens the end of driving permission position (EOA) to the end of the track circuit information permission, until LRBG update or code sequence changes to send R132 M132 (request for driving permission), received a new MA[5].

At present, there are two main ways for RBCs produced by domestic manufacturers to send MAs: 1. Each time a MA starting at 0 is sent, that is, this MA generally contains the line information that has been sent before. 2. In addition to receiving M132 (requesting MA) sent by the vehicle, when it is informed that the line data has been deleted, it will send the MA starting at 0. In other cases, the MA is sent by stitching, that is, the starting point of this MA is the last one sent. At the end of MA, the line information that has been sent before will not be sent repeatedly[6]. According to the above description, in order to verify whether the equipment of each manufacturer can complete the interconnection and interoperability, the working logic of the autonomous CTCS-400C vehicle-mounted equipment and the domestic RBC (the second RBC sending MA method above) is selected for analysis and modeling.

2.2. Security requirements for code sequence verification scenarios

To complete the code sequence verification scenario, it is necessary to cooperate closely with the vehicle equipment, RBC, ground transponder, track circuit, and driver to complete together. The vehicle-mounted device should receive the ground transponder information, establish a connection with RBC, send a message to it and receive the driving authorization (MA) sent by RBC, and receive the track circuit code, and start the code sequence verification when the code sequence is L3 and below. The entire process requires the following actions: the train establishes a connection with the RBC, receives the position information of the ground transponder, sends the position report, obtains the driving permission, and performs the code sequence verification[7]. The security of the system is used to describe the things that may or may not happen to the system. Popularly speaking, it means "bad things will never happen." The functionality of the system is used to show that something must happen to the system, which means "good things. It will happen after all." Combining the literature and the above workflow, the following code sequence verification scenario security function attributes, functionality and time requirements are proposed as verification targets[8].

2.2.1. Safety function requirements
(1) The system does not deadlock;
(2) Only one of EVC_wait_spliceMA or EVC_wait_newMA in the EVC time automaton can be reached within the same time;
(3) After switching from C3 to C2, the train speed is not higher than 250km / h;
(4) When the equipment is normal, the code sequence can be verified.
2.2.2. Functional requirements
(1) Within the time $T_{\text{waitack}}$, RBC does not receive the confirmation from the vehicle to the MA, and then resends the MA;
    (2) If the vehicle sends an MA request to the RBC, the RBC must receive the vehicle MA request or the train position report.

2.2.3. Time characteristics requirements
(1) The connection time between vehicle and RBC is not more than 10s;
    (2) The time for judging the wireless timeout is 20s.

2.3. Code sequence verification scene model based on time automata
The scene is selected as the process of the train driving from L4 code to L3 code section. Some details of the process are simplified as follows: the default average slope of the line is less than 0 ° and greater than minus 10 °, then the average length of the block is 2100 meters; simplified The process of establishing and breaking the train and RBC; assuming that each block has only one set of transponders, and the transponder is located at the beginning of the block, a module can be used to describe the transponder (LRBG) and the track circuit (code sequence) Function.

Through the analysis of the workflow of the above scenario, the vehicle-mounted equipment (EVC), wireless block center (RBC), track circuit (TCR) and driver (Driver) are modeled for the model subsystem. The model structure is shown in Figure 2

![Diagram](image)

Fig.2 Structure diagram of Code sequence check scenarios model

(1) The modeling objects are selected as EVC, RBC, Driver, and TCR, and the code sequence verification model of each object is a time automaton, which is sequentially denoted as TAEVC, TARBC, TACCR, and TADriver, then the code sequence verification workflow model The product of the above four time automaton models is written as [9]:

$$TA = TAEVC \ || \ TARBC \ || \ TATCR \ || \ TADriver$$

(2) The "functions" in the extraction technical specifications are as follows: send, wait, receive, validity judgment, shortening, repeated prompting, timeout judgment, prompting, confirmation ...

The above "function" corresponds to the position in the time automaton model. Set location collection:

$S_0 = \{\text{idle}\}$,
And S = SEVCYRCYSRBCYSTCRYSDriver.

among them,

SEVC = \{idle, EVC\_wait\_RBC, EVC\_C2, EVC\_sendposition, EVC\_MAcheck, EVC\_shortenMA, 
EVC\_updateMA, L\},

SRBC = \{idle, RBC\_ACK, RBC\_wait\_position, RBC\_sendMA, RBC\_receivedM132, 
RBC\_received\_newposition, L\},

STCR = \{idle, TCR\_keepcode\},

SEVC = \{idle, PrePress\_button, Error\}.

(3) The “time characteristics” of the technical specifications are extracted as follows: within 10 seconds, at intervals of 6 seconds, and over 20 seconds, use time automata to describe the above characteristics, and set the event collection:\[10]:

X = \{T0, T1, T2\}

(4) The "communications" of technical specifications are extracted as follows: call, response, received code sequence, location report, driving permission, LRBG update, retransmission, wireless timeout, communication interruption, etc. the above "communication" content takes time automata

The event A protocol is used to describe the communication between members' time automata. Set the event set:

A1 = \{connect, receive, M132, M3, M136, new\_MA\_From0, new\_splice\_MA, 
refuse\_new\_MA, resend\_MA, L\},

A2 = \{L3, finish\},

A3 = \{remind\_driver, Reconfirm, confirm\_C2, noconfirm\_C2, reconfirm\}. Among them, A1 implements synchronous information interaction between EVC and RBC, A2 implements synchronous information interaction between TCR and EVC, and A3 implements synchronous information interaction between driver and EVC.

(5) The "environment" of the extraction technical specifications is as follows: after receiving, timeout, interval, ... "environment" describes the constraints for the transition of each state of the time automaton model. The following takes the safety function attribute requirements, functionality, and time requirements described in Section 2.2 as examples to illustrate the relationship between member automata. The EVC sends an event connect (connect \(\in\) A1) to the RBC to establish a communication command. According to the definition of time motivation, TAEVC switches.

\{idle, connect, \{T0\}, EVC\_wait\_RBC\},

And reset the clock T0 to 0 to start timing.

Among them, \{idle, EVC\_wait\_RBC\} \subseteq SEVC.

At the same time, RBC received the event connect, TARBC converted \{idle, connect ,, 
RBC\_ACK\},

Among them, \{idle, RBC\_ACK\} \subseteq SRBC.

As time T0 elapses, RBC returns a confirmation result to EVC (receive \(\in\) A1) within 10S. At this time, RBC and EVC are converted simultaneously.

\{RBC\_ACK, receive ,, RBC\_wait\_position\}

with

\{EVC\_wait\_RBC, receive, T0 <= 10, EVC\_receive\_ACK\}. Among them, \{EVC\_wait\_RBC, EVC\_C2\} \subseteq SEVC,

\{RBC\_ACK, receive ,, RBC\_wait\_position\} \subseteq SRBC. If there is an execution sequence r1, r2, such that

\{idle, v\} r1 (EVC\_wait\_RBC, v) \quad (2)

And

\{idle, v\} r2 (RBC\_ACK, v) \quad (3)

Established. Among them, v \in Tx. Then, it can be explained that EVC issued a call RBC command, and at the same time, RBC also received a call command. If there are execution sequences r3, r4, such that
\{RBC\_ACK, v\} r3 (RBC\_wait\_position, v) \quad (4)

And

\{EVC\_wait\_RBC, v\} r4 (EVC\_receive\_ACK, v) \quad (5)

Established. Among them, v ∈ Tx.

Then, it can be explained that RBC returned the confirmation result to EVC within 10S, and at the same time, EVC also received the confirmation message. Whether there is an execution sequence r1, r2, r3, r4, so that the formula is established, it belongs to the verification work of the following automatic verification tool UPPAAL.

3. System modeling and simulation

3.1. Time Automaton and UPPAAL

Time automaton is a formal modeling method based on time period, time constraints and time interpretation. UPPAAL verification tool was jointly developed by Aalborg University and Uppsala University in 1995\cite{11}. It controls a limited number of control structures and numerical clocks through simulation and uses integral Type variables to verify the safety and limited activity of the system. The UPPAA user interface includes three parts: a system editor, a simulator, and a verifier. The system editor is used to create and edit the system to be analyzed; the simulator is used to check whether the built system model is correct; verifier checks the clock constraints and limiting properties by quickly searching the state space of the system model\cite{12}. The UPPAAL verification tool uses BNF grammar. The meaning of this grammar is shown in Table 1.

| BNF syntax       | Specific meaning                                                                 |
|------------------|----------------------------------------------------------------------------------|
| E <>             | E <> is true if and only if there is a path s0 → s1 → …… → sn, so that p is satisfied in a certain state si in the path state sequence |
| E[]p             | E [] p is true if and only if there is a path s0 → s1 → …… → sn, so that p is satisfied in all states si |
| A<>p             | Equivalent to not E [] not p, A <> p is true means that all possible path state sequences will eventually pass through a state, which satisfies the expression p |
| A[]p             | Equivalent to not E <> not p, A [] p being true means that every reachable state in all possible path state sequences satisfies the expression p |
| p→q             | Equivalent to A [] p, A [] p is true means that when the expression p occurs, the expression q will eventually occur |

3.2. scene model diagram

According to the scene process described in Sections 2 and 1, refer to the literature, and according to the key information extracted in Sections 2 and 2, establish the code sequence to verify the scene driver, TCR, EVC, RBC models are shown in Figures 3, 4, 5, and 6.

Fig.3 Driver module model of code sequence verification scenario
At this point, the modeling work of the code sequence verification scenario is completed, and the model is guaranteed to reflect the technical specifications as much as possible, which lays the foundation for the verification of the model below. After the temporary speed limit system model is correctly established, the system characteristics reflected by the model need to be verified [13]. This paper uses UPPAAL automatic verification tools to verify the characteristics of the system proposed in Section 2.3. Before verification, before the system model verification, the UPPAAL verification tool simulator needs to be simulated to check the model for syntax errors. The time sequence diagram of the information interaction generated by the random model during the simulation process is shown in Figure 8. It is a reflection of the workflow of the interaction between devices in the code sequence verification scenario. Ensure the accuracy and feasibility of verification results.
4. Verification of system model

For the verification of the security function attributes of the code sequence verification scenario, it can be attributed to the reachability analysis of the time-directed automaton. The functionality and time characteristics of the system are guaranteed by the invariance of the position and the constraints of the conversion [14].

4.1. System security function attributes

1) A [] not deadlock, the system is not deadlocked, and the verification is passed.
2) A <> (EVC.EVC_wait_spliceMA) + (EVC.EVC_wait_newMA) <= 1, it is required that only one of the EVC time automaton EVC_wait_spliceMA or EVC_wait_newMA can be reached within the same time, and the verification is passed.
3) A <> ((EVC.EVC_wait_driver imply EVC.EVC_changeC2) and (speed <= 250)), after switching from C3 to C2, the train speed is not higher than 250km / h, the verification is passed.
4) E <> (EVC.idle imply EVC.EVC_MAcheck), when the devices are normal, the code sequence can be verified and the verification is passed.

4.2. System time characteristics requirements

1) A <> ((EVC.EVC_wait_RBCimply EVC.EVC_receive_ack) and (T0 <= 10)), the connection time between vehicle and RBC is not more than 10S, the verification is passed.
2) E <> ((RBC.RBC_MA_resend imply RBC.RBC_wireless_timeout) and (T2> 20)), judging that the wireless timeout time is 20S, the verification is passed.

4.3. System functional requirements

1) E <> ((RBC.RBC_NOTreceived_ACKimplyRBC.RBC_MA_resend) and (T1> Twaitack), within the time Twaitack, RBC does not receive the confirmation from the vehicle to the MA, and then resends the MA. The verification passes.

Each of the above properties has been verified one by one, and the analysis shows that the system meets the actual requirements of CTCS-3 level code sequence verification scene security function attributes, timeliness and functionality.

5. Conclusion

This paper uses the formal method of time automata to model the code sequence verification scenario, and uses UPPAAL to establish a model of the entire code sequence verification scenario information interaction system, elaborating the information interaction and state transition process in the model in detail, and
according to the process described in the technical specifications, establishes a network model of timed automata, and uses UPPAAL to verify the model. The security, timeliness and functionality of the CTCS-3 code sequence verification scenario were successfully verified, which ensured the smooth operation of the system and provided an important basis for the further development of the system.

Acknowledgments
This publication is based upon work supported by the National Natural Science Foundation Regional Science Foundation Project under Project approval number: N2019G017; GSITA201910. The authors would like to thank anonymous referees whose comments and suggestions have improved the content and presentation of this work.

References
[1] China Railway Corporation. Temporary technical conditions for autonomous C3-level train-controlled onboard equipment: TJ/DW 205-018 [S]. Beijing: China Railway Corporation, 2018.
[2] China Railway Corporation. CTCS-3 level ATP driving permit combined with track circuit information temporary technical conditions: TJ/DW 200-2018 [S]. Beijing: China Railway Corporation, 2018.
[3] Cheng Yajing. Modeling and verification of complex scene of autonomous CTCS-3 train control system [D]. Beijing: Beijing Jiaotong University, 2018.
[4] An Yue. Formal modeling and analysis of RBC system based on UML and TA [D]. Lanzhou: Lanzhou Jiaotong University, 2016: 7-37.
[5] Zhang Ailing. Formal modeling and analysis of CTCS-3 train control system RBC driving permit generation [D]. Lanzhou Jiaotong University, 2012.
[6] Larsen K G, Mikucionis M, Nielsen B. Online Testing of Real-time System Using UPPAAL, Formal Approaches to Software Testing-4th International Workshop [J]. Lecture Notes in Computer Science: (S0302-9743), 2005, 3395: 79-94.
[7] Lv Jidong, Tang Tao. Modeling and verification of real-time operation scenarios of high-speed railway train control system [J]. Journal of the China Railway Society, 2011, 33 (6): 54-61.
[8] Cao Yuan. Research on Formal Modeling and Verification Method of High-Speed Train Operation Control System [D]. Beijing: Beijing Jiaotong University, 2011, 3-60.
[9] Kang Renwei. Research on Modeling Method and Verification of CTCS-3 Train Control System Based on Time Automata [D]. Beijing Jiaotong University, 2013.
[10] Cao Jiayun. Modeling and verification of CTCS-3 level train-controlled on-board equipment based on time automata [D]. Southwest Jiaotong University, 2008.
[11] Geng Peng. Modeling and verification of driving permission module of RBC system based on safe state machine [D]. Beijing Jiaotong University, 2009.
[12] Hessel A, Larsen K G, Mikucionis M. Testing real-time systems using UPPAAL [J]. Lecture Notes in Computer Science, 2008, 4949: 77-117.
[13] Lu Jidong, Li Kaicheng, Tang Tao, et al. Formal modeling and verification method of high-speed railway train control system based on hybrid communication sequence process [J]. China Railway Science, 2012, 33 (5): 93-99.
[14] Guo Hua, Zhuang Lei, Zhang Xiyong. UPPAAL is a tool suitable for automatic verification of real-time systems [J]. Microcomputer Information, 2006, 22 (5): 52-55.