Towards A Test Paths Generation Method for CTCS Level Transition

LI Yao[1,a], ZHANG Xiaoxia[1], ZHANG Yadong[2], GUO Jin[2] and GAO Hao[2]

[1]*School of Optoelectronic Science and Engineering, University of Electronic Science and Technology of China, SiChuan, China*
[2]*School of Information Science and Technology, Southwest Jiaotong University, SiChuan, China*

Abstract. Test case is an important basis for correctness and safety verification of Chinese Train Control System (CTCS). Focusing on the test cases generation method of UPPAAL which is widely used in CTCS testing activity, the problems are analyzed, and an improved test cases generation method for CTCS is proposed. First, the process and characteristics of UPPAAL test cases generation method are analyzed; then the test requirements of CTCS are studied, and a test cases generation method based on UPPAAL query file is proposed. Finally, taking the level transition function of CTCS as an example, test cases are generated by the proposed method, which shows that this method can meet the test requirements of CTCS.

1 Introduction

China's high-speed rail business mileage reached 29,000 kilometers, more than two-thirds of the world's total. In order to guarantee the safety of train operation, improve the transportation efficiency, the Chinese Train Control System (CTCS) is proposed by railway department based on Chinese national conditions. CTCS is a typical safety critical system with a safety requirement of SIL 4, whose failure may cause rear end collision, derailment, and other accidents, resulting in casualties and major property losses. Ensuring the safety and correctness of CTCS is of great significance for the safe operation of Chinese railway.

UPPAAL is a tool supporting modeling, simulation, formal verification, and test cases generation of Timed Automata, which has been widely used in CTCS. This paper studies the problems of UPPAAL test cases generation method in CTCS testing effort, and an improved method for CTCS based on UPPAAL query file is proposed. Level transition is an important safety function in CTCS, and this paper takes this function as an example to show the effectiveness of the proposed method in CTCS test cases generation effort.

2 RELATED WORK

As an important safety function of CTCS, level transition is causally related to train operation safety and transportation efficiency, whose simulation, verification, and test work have been studied widely. In paper12, the level transition process of high-speed railway train control system is modeled and verified by Time Automata. Paper3 and paper4 analyze the level transition function using methods of UML, MSC and UPPAAL respectively. Paper5 studies the level transition function in case of balise failure. Paper6 analyzes the level transition logic, and designs test cases based on scene method . Paper7 researches mutation testing method based on Timed Automata and designs test cases for level transition function.

Timed Automata is an important formal method and has been widely used in CTCS[8910]. Paper111213 studies the test cases generation method based on Timed Automata for CTCS. UPPAAL is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of Timed Automata, and especially provides Yggdrasil to generate test cases automatically from the Time Automata model. Due to these efforts, Time Automata has been paid more attention in the testing activities of CTCS.

This paper first analyzes the problems of UPPAAL test traces generation, then analyzes the testing characteristics of CTCS, and puts forward a test paths generation method for CTSC based on UPPAAL query file test technique; finally, taking level transition function as an example to show the effectiveness of this method.

3 PRELIMINARIES

3.1 Introduction to TA

Timed Automata[14] (TA) is a formal modeling method for real-time system, which has a rigorous mathematical theoretical basis and plays an important role in the specification and formal analysis of real-time systems. TA adds the clock constraint mechanism based on the traditional finite state machine and makes the behavior of the traditional finite state machine with the characteristics of time.

[a] Corresponding author: liyao_678@163.com

A Timed Automata is a tuple $\langle L, L_0, \Sigma, X, I, E \rangle$, where[14]:

- L is a finite set of locations.
- $L_0 \subseteq L$ is a set of initial locations,
- Σ is a finite set of labels,
- X is a finite set of clocks,
- I is a mapping that labels each location l with some clock constraint in $\Pi(X)$,
- $E \subseteq L \times \Sigma \times 2^X \times \Pi(X) \times L$ is a set of transitions.

A transition $\langle s, a, \varphi, \lambda, s' \rangle$ represents an edge from location l to location l' on symbol a, φ is a clock constraint over X that specifies when the transition is enabled, and the set $\lambda \subseteq X$ gives the clocks to be reset of this transition.

3.2 UPPAAL Test Cases Generation

UPPAAL is a TA integrated tool environment developed by Uppsala University and Aalborg University, which extends the TA and uses on the fly technology to analyze the reachability and verify the safety and liveness of TA model.

UPPAAL provides Yggdrasil to generate test traces from TA model and translates them into test cases based on test code entered into the model. UPPAAL generates traces in three phases: query file, depth search, and single step.

(1) The first phase looks through the currently loaded query file in the verifier for reachability queries. Each reachability query is executed, and the resulting trace is used as a test case.

(2) The second phase does a random depth first search of the specified number of steps, and the resulting trace is used as a test case. This process is repeated until the newly generated trace does not add new coverage over the previous traces.

(3) The third phase tries to cover any remaining edges not covered by the previous phases. This is done by making a reachability query for each uncovered edge.

The three phases are executed in order, and each phase tries to add to the coverage achieved by previous phases.

4 TEST PATHS GENERATION METHOD

4.1 Problem of UPPAAL

UPPAAL uses a randomized algorithm to search test traces, resulting in high repeatability and uncertainty. For CTCS testing activity, the testing coverage criteria of path coverage is required, but UPPAAL generates traces for edge coverage, fails to meet the test requirement of CTCS.

(1) High Repeatability

In the second phase of UPPAAL test traces generation process, UPPAAL generates test traces by random depth first search algorithm. The random algorithm leads to uncertainty and high repeatability.

Taking figure 1 as an example, four experiments results of depth search are shown in table 1. Among the four depth search results, the maximum number of traces is 26, and the minimum number is 17, and only experiment 3 achieves 100% edge coverage.

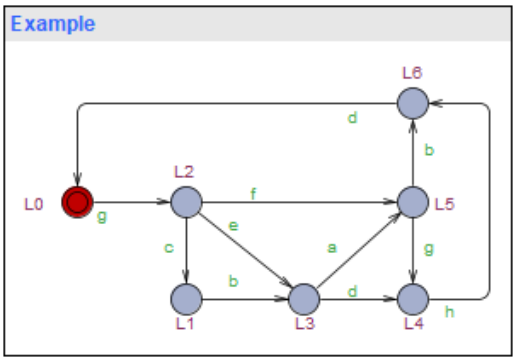

Figure 1. Example Model TA_{Example}

Table 1. Depth Search Experiments

	Trace NO.	Trace Coverage	Total Coverage	Total Traces
1	1	6/11	10/11	25
	2	9/11		
	3	10/11		
2	1	9/11	10/11	26
	2	8/11		
	3	9/11		
3	1	9/11	11/11	18
	2	9/11		
4	1	10/11	10/11	17
	2	7/11		

It can be seen from the table that the test traces generated by UPPAAL are uncertain and have high repeatability, resulting in long testing time and low test efficiency.

(2) Not Path Coverage

UPPAAL generates test traces for edge coverage, but the test of CTCS should meet the requirements of path coverage. Edge coverage cannot guarantee the adequacy and completeness of CTCS test requirements. Taking TA_{Example} as an example, a test trace is as follows:

L0→L2→L1→L3→L5→L4→L6→L0→L2→L5→L6→L0→L2→L1→L3→L4→L6→L0→L2→L3→L5→L6.

The test trace covers all edges of the model, and 4 test paths, but TAExample includes 8 test paths. UPPAAL have the problem of incomplete path coverage in CTCS testing activity.

4.2 Test Cases Generation Method

The functional model M of CTCS can be described as:

$M = M_{SUT} \parallel M_{Env}$,

where:

M_{SUT} is the TA model of system under test (SUT),

M_{Env} is the TA model of environment for M_{SUT}.

For CTCS, the test requirements require that all paths from initial location to final location should be covered, that is, all paths of M_{SUT} should be covered.

Focusing on this test requirement, this paper proposes a test path generation algorithm with UPPAAL query file, which aims to cover all model paths and reduce the repetition at the same time.

(1) Test Model

The test model $M_{SUT}^{/}$ is formed through adding tag variables on the basis of M_{SUT}. $M_{SUT}^{/}$ does not change the function of M_{SUT}, but only add an ability to distinguish each path by values of tag variables in test paths generation process.

The test model of $TA_{Example}$ in figure 1 is shown in Figure 2. The variables of fork 1, fork 2 and fork 3 mark the branches of TAExample, and each test path can be defined by the value of these variables. For example, (fork1==3) and (fork3==1) indicates a path through L0, L2, L5 and L6.

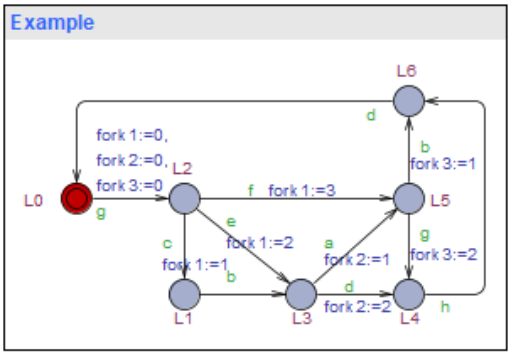

Figure 2. Test Model of $TA_{Example}$

(2) Test Path Generation Algorithm

The test path generation algorithm is based on the test traces generation method of UPPAAL query file. Let P denotes the path set of M_{SUT} that CTCS test requirements required, t denotes a test path generated by the current iteration of the algorithm, R denotes the total paths generated by the algorithm, q denotes the final location needed to be reached of the model according to the test requirements of CTCS. The test path generation algorithm is as follows:

```
R = Ø
while P ≠ Ø
  generates trace(s) t using E<> (q and !R)
  R = R ∪ t   /*add the trace to the result set*/
  P = P / t   /*delete the trace from P*/
```

E<> (q and !R) is a query file of UPPAAL and indicates the path in the test requirements, excluding the path in R. At the beginning of the algorithm, R is an empty set. Each iteration of the algorithm generates one or more test paths, and adds the test paths to R. The algorithm iterates until R is empty, i.e. all test requirement paths are found, which can be translated into test cases by UPPAAL test code.

5. INTRODUCTION TO LEVEL TRANSITION

CTCS includes 5 application levels of CTCS0 - CTCS4, and multiple application levels can be realized on the same line to meet the requirements of different line speeds. Different application levels are switched by the level transition function.

5.1 C3/C2 Level Transition Analysis

The train is supposed to switch its run level from CTCS3 to CTCS2 (C3/C2) when it enters C2 railway line from C3 line, or the communication between train and RBC (Radio Block Center) is interrupted. It is an important safety function that may cause traffic interruption, even traffic accidents, if the train does not switch correctly at the expected position. It is of great significance to verify the safety and correctness of the level transition function logic to ensure the train operation safety.

In order to ensure the stability of the level transition progress, two balises named LTA and LTO are set on the wayside, which are used to provide level transition announcement and execution information respectively. The schematic diagram of C3/C2 level transition is shown in Figure 3.

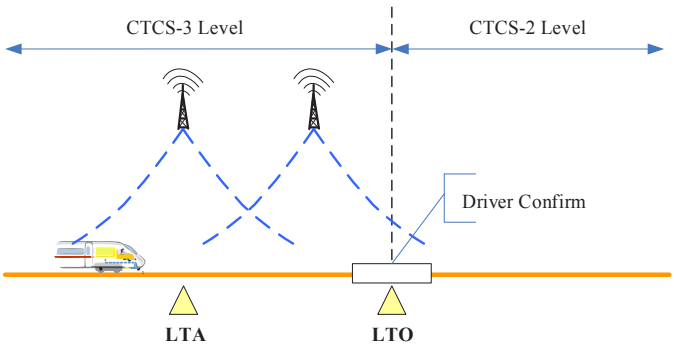

Figure 3. C3/C2 Level Transition Schematic Diagram

5.2 Function of C3/C2 level transition

The main workflow of C3/C2 transition is illustrated as figure 4, including four key steps:

(1) When the train passes LTA, ATP (Automatic Train Protection) reports train position to RBC. After receiving this report, RBC sends a level transition command from C3 level to C2 level to ATP.

(2) ATP requests the driver to confirm this level transition command, and the driver should confirm it within 5 seconds.

(3) When the train passes the LTO point, ATP will switch to C2 level and reports the train position to RBC.

(4) After receiving the report, RBC commands ATP to disconnect from RBC. ATP will close the connection

and send the termination communication meeting information to RBC.

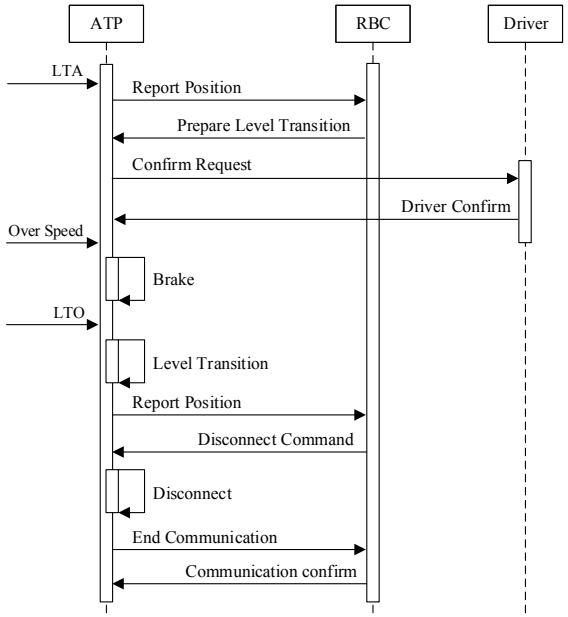

Figure 4. Sequence Diagram of C3/C2 Level Transition

5.3 TA model of C3/C2 level transition

The TA model of C3/C2 level transition is shown in figure 5, including 5 parts: ATP, RBC, Driver, Balise and Train:

$M_{C3/2} = M_{ATP} \parallel M_{RBC} \parallel M_{Driver} \parallel M_{Train} \parallel M_{Balise}$.

For simplicity, this chapter takes M_{ATP} as the SUT, then $M_{RBC} \parallel M_{Driver} \parallel M_{Train} \parallel M_{Balise}$ is the environment model of M_{ATP}. The main channel is shown in table 2.

Table 2. Main channel of $MC_{3/2}$

Channel	Meaning
TrPassLTO	Train Pass LTO
DvrCfmRst	Driver Confirm Request
CloseCmd	Close Command
TrCloseLink	Train close link

5.4 Test path of M_{ATP}

Using the method proposed in chapter 4, the test paths generated from M_{ATP} are shown in table 3. For simplicity, only the fork part of every path is described.

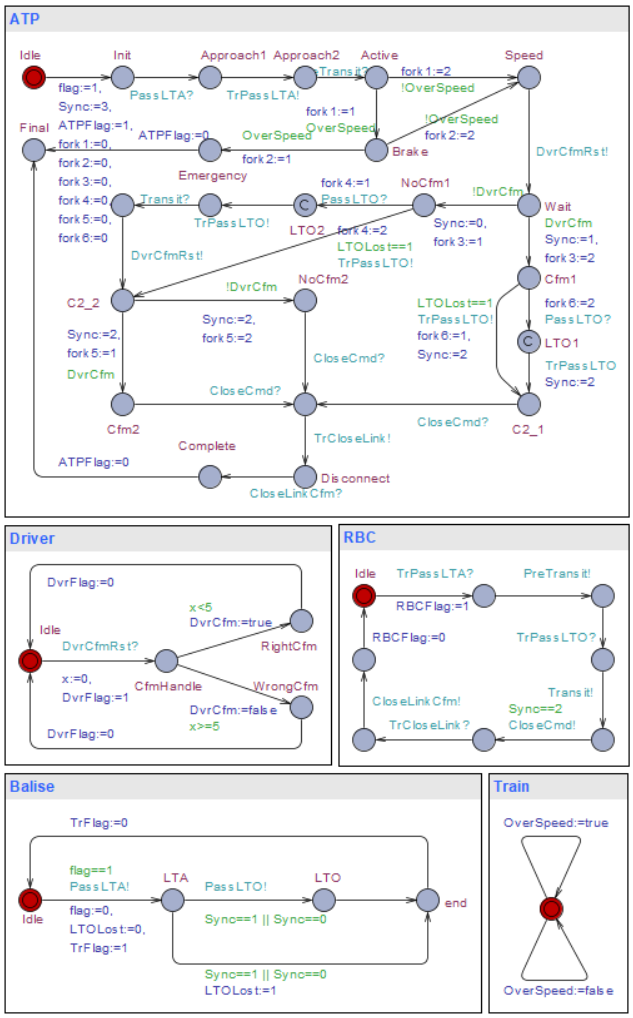

Figure 5. TA model of C3/C2 level transition

Table 3. Test Paths of $TA_{C3/2}$

NO.	Test Path
1	Active, Speed, Wait, NoCfm1, C2_2, NoCfm2
2	Active, Speed, Wait, NoCfm1, C2_2, Cfm2
3	Active, Speed, Wait, NoCfm1, LTO2, C2_2, Cfm2
4	Active, Speed, Wait, NoCfm1, LTO2, C2_2, NoCfm2
5	Active, Speed, Wait, Cfm1, C2_1
6	Active, Speed, Wait, Cfm1, LTO1
7	Active, Brake, Speed, Wait, NoCfm1, C2_2, NoCfm2
8	Active, Brake, Speed, Wait, NoCfm1, LTO2, C2_2, NoCfm2
9	Active, Brake, Speed, Wait, NoCfm1, LTO2, C2_2, Cfm2
10	Active, Brake, Speed, Wait, NoCfm1, C2_2, Cfm2
11	Active, Brake, Speed, Wait, Cfm1, C2_1
12	Active, Brake, Speed, Wait, Cfm1, LTO1
13	Active, Brake, Emergency

The proposed method generates 13 test paths, and 100% covers the paths and edges of M_{ATP}. The result

shows that the proposed method meets the requirements of CTCS testing activity.

6 CONCLUSION

This paper analyzes the problems of UPPAAL test traces generation process in CTCS testing activity. Focusing on the test requirements of CTCS, a test paths generation method is proposed based on the query file test case generation method of UPPAAL. Taking level transition as an example, TA model is established, and test paths are generated by this method.

The method proposed in this paper has been applied in the system test of TCC (Train Control Center) and ATP system of CTCS. The practice shows that the method can improve the test completeness and test efficiency and meet the test requirements of CTCS.

7 ACKNOWLEDGMENTS

This work is partially supported by the China State Railway Group Co., Ltd. under Grant N2018G062. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

Reference

1. Kang Renwei, WANG Junfeng, LU Jidong. UPPAAL-based modeling and verification of level transition process of high-speed railway train control system. Journal of Beijing Jiaotong University, 2012(6):63-67.
2. Ye Anjun. Modeling and Verification of ATP Level Transition Process Based on Timed Automata. Urban Mass Transit,2019,22(07):27-32+37.
3. Hu Xuelian. Modeling and Verification of Level Transition Scene in CTCS-3 Level Train Control System Based on UML and UPPAAL. Lanzhou Jiaotong University, 2015.
4. Hu Xuelian, Tao Caixia. Formal Verification of Level Transition Process in Train Control System Based On MSC and UPPAAL. Railway Standard Design, 2015, 000(002):122-127.
5. Wang Yuanpeng, HU Xiaohui, CHEN Yong, et al. Modeling and simulation of transponder failure due to CTCS level conversion. Computer Engineering and Applications, 2016, 052(008):234-239.
6. Dou Lei，ZHANG Ya-dong，LI Yao, et al. Design of Test Cases for Level Transition Function of Train Control System Based on Scene Method. Railway Standard Design, 063(007):141-145,152.
7. Li Teng. The Research on Mutation Testing Method of Chinese Train Control System Level 3 Based on Timed Automata. Beijing Jiaotong University, 2016.
8. Yuan Lei, WANG Junfeng, KANG Renwei, et al. Modeling and Verification of Temporary Speed Restriction of CTC-S3 Train Control System. Journal of Southwest Jiaotong University, 2013, 048(004):708-714.
9. Lv Ji-dong, TANG Tao, JIA Hao. Modeling and Verification of Radio Block Center of CTCS-3 Train Control System for Dedicated Passengers Lines. Journal of The China Railway Society, 2010(6):34-42.
10. Shi Tingrui. Modeling Analysis and Implementation of RBC Handover Based on Timed Automata. Beijing Jiaotong University,2019.
11. Guo Haonan, Lv Jidong, CHAI Ming, et al. Research on Verification of CBTC Onboard ATO Functions Based on Online Conformance Testing Theory,2020,42(03):93-103.
12. T. Wang, J. Lv, B. Wei, T. Tang and W. Shangguan, Test Suite Generation for CTCS-3 Train Control System Based On TAIO and Mutation Theory, 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, 1950-1955.
13. Z. Hu et al., Fault Diagnosis of the On-board Equipment in CTCS-3 Based on Timed Automata and Mutation Theory, 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, 1013-1018.
14. Alur R, Dill D L. A theory of timed automata. Theoretical Computer Science, 1994, 126(2):183-235.