SCADE-BASED AUTONOMOUS ATP LEVEL TRANSFORM TEST CASE GENERATION METHOD

HUISI XU, YADONG ZHANG*, JIN GUO, YAO LI, HAO LAN

The School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China
**Corresponding author's email: ydzhang@swjtu.edu.cn*

Abstract: *At present, most of the tests for autonomous ATP systems rely on manual writing of test cases. This method relies on the experience of testers, and the resulting test cases are less correct and take longer. This paper proposes a method based on Safety Critical Application Development Environment (SCADE) to generate test cases, taking the autonomous ATP level transform from CTCS-3 level to CTCS-2 level as an example, using SCADE to accurately model the function, and observing the coverage information of the node through Model Test Coverage (MTC). Model input variables are modified to adjust model coverage and automatically generate test cases.*

Keywords: *SCADE; Autonomous ATP; Level Transform; Test Case*

I. INTRODUCTION

With the rapid development of China's high-speed railway industry, as of the end of 2018, China's high-speed railway operating mileage exceeded 29,000 kilometers, accounting for more than two-thirds of the world's high-speed railway operating mileage, exceeding the sum of other countries. In response to China's "high-speed railway going out" and "the Belt and Road" development strategy, China Railway Corporation carried out research on the independent control technology of the train control system, and developed a number of safety key equipment with independent intellectual property rights [1,2]. Among them, the Automatic Train Protection (ATP) system is a key part to ensure the safety of trains and improve transportation efficiency. The ATP system has the safety requirement of SIL 4, and once it fails, it may cause train accidents such as train rear-end and derailment, resulting in casualties and major property losses. Therefore, testing the ATP system is of great significance for the safe operation of the high-speed railway.

At present, most of the tests for ATP systems rely on manual writing test cases. This approach relies on the experience of the tester and lacks completeness and correctness. For logic complex functions, it is difficult to manually write test cases, which is easy to cause omissions [3]. Because SCADE has a strict mathematical theoretical basis, it can express complex systems accurately. The model test coverage analysis tool MTC can measure the coverage of the test cases and give hints to the uncovered parts for the tester to modify the test cases [4].

This paper studies the SCADE-based test case generation method, and uses the

autonomous ATP level transform function as the object to generate test cases by using the test case coverage tool MTC.

The rest of the paper is organized as follows. Section II provides an overview of SCADE. Section III describes the autonomous ATP level transform logic and analyzes the CTCS-3 level to CTCS-2 level transform process. Section IV models the autonomous ATP level transform by SCADE. Section V generates test cases for the CTCS-3 level to CTCS-2 level transform. Section VI summarizes characteristics of the methods.

II. OVERVIEW OF SCADE

SCADE is a model-driven, high-security application development software with a rigorous mathematical theory foundation that provides a model-based embedded development solution for security-critical systems and high-security software. It has been widely used in aviation, railway, nuclear power and other fields with high security [5].

1. Safe State Machine Modeling

The safe state machine is a modeling language based on Synchronization Hypothesis. It is mainly used in the research of discrete systems, with the characteristics of hierarchical structure, priority preemption and concurrency control. The attributes of the safe state machine are: status, signal, and migration. The state is usually used to describe the situation in which the system is located. Within one cycle, one and only one state is active. And a state can have multiple transitions. When a state is activated and multiple migration conditions are true, the state machine only triggers the migration with the highest priority. This is the priority feature of the safe state machine, which guarantees the uniqueness and certainty of the state of the system.

2. Model Simulation

Model simulation is mainly used to test whether the model meets system functions. SCADE implements model simulation verification technology through SCADE simulator. The simulation object can be the whole system or a module. Execute the model by inputting the signal and see if the output meets the system requirements. The SCADE simulator can write scene files for scene simulation. At the same time, the scene file can be saved and played back without manual input of variable values, which reduces the workload of the tester and improves the test efficiency.

3. Model Test Coverage

In addition to model simulation, model test coverage is needed to determine the completeness of model simulation. Before the model coverage analysis, the corresponding criteria should be selected to analyze the coverage of the simulation scenario. Modified Condition/Decision Coverage (MC/DC) is a practical software structure coverage test criterion that has been widely used in software verification and testing processes, requiring each condition of a decision to independently affect the outcome of the decision. MTC can load existing test cases, simulate the coverage information of all nodes, and analyze the results. Testers can adjust test cases based on uncovered conditions. Model Test Coverage process is shown in figure 1.

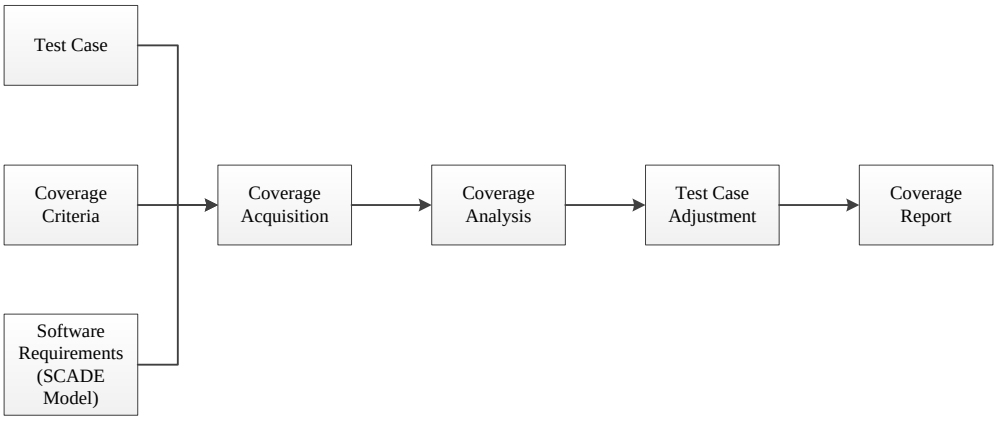


Figure 1. *Model Test Coverage Process*

III. AUTONOMOUS ATP LEVEL TRANSFORM

According to the actual needs of China, the train control system is divided into CTCS-0, CTCS-1, CTCS-2, CTCS-3 and CTCS-4. ATP is mainly used in CTCS-2 and CTCS-3. In autonomous ATP, level transform is one of the important operational scenarios. It is of great significance to ensure train safety during train level transform [6].

1. Level Transform Function

The level transform includes two scenarios: transform from CTCS-2 to CTCS-3 (C2/C3) and transform from CTCS-3 to CTCS-2 (C3/C2). This paper describes the transform from CTCS-3 to CTCS-2 as an example.

When the fixed point performs transform from CTCS-3 to CTCS-2, after receiving the level transform command, the CTCS-3 module shall send the level transform position information to the CTCS-2 module, and the CTCS-2 module reports the target speed of the level transform point to the CTCS-3 module. When the front end of the train crosses the transform execution point, the CTCS-3 module commands the CTCS-2 module to enter the foreground working state. When the CTCS-2 module is in the foreground, the CTCS-3 module is responsible for the supervision of the onboard equipment system.

The CTCS-3 level transform to CTCS-2 level mainly includes: level transform notice and level transform execution. When the onboard equipment performs level transform at a fixed point, the onboard equipment automatically switches to a new level without driver confirmation. If the onboard equipment fails to convert to the CTCS-2 level from CTCS-3 level at the transition point, the brake command should be output and stop. The operation scenario is shown in figure 2.

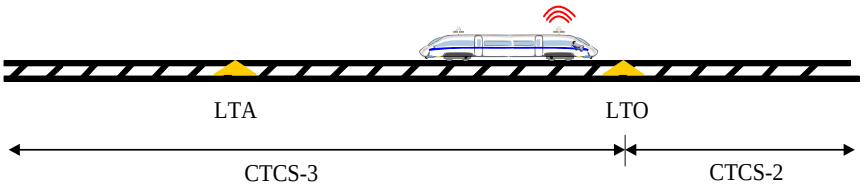


Figure 2. *Operational Scenario*

2. Level Transform Analysis

The sequence diagram of CTCS-3 level transform to CTCS-2 level is shown in figure 3.

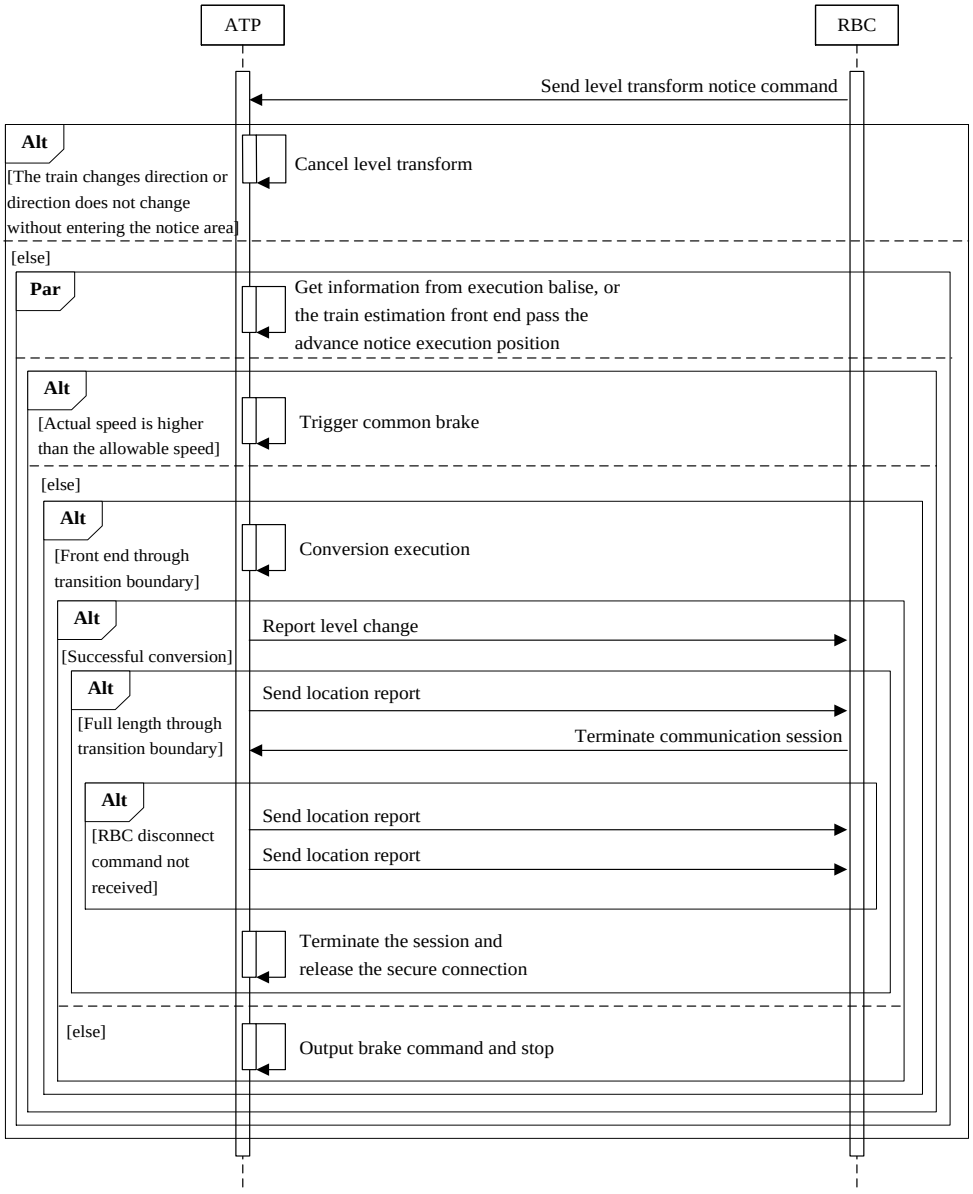


Figure 3. *CTCS-3 Level to CTCS-2 Level Transform Sequence Diagram*

A. Level Transform Notice

When the train needs to be converted from CTCS-3 to CTCS-2, the RBC shall send a level transition notice command to the ATP when the train front is at a certain distance from the transition boundary. If the train orientation changes, or the orientation does not change without entering the preview area, the level transform is cancelled and the train maintains the CTCS-3 level operation.

B. Level Transform Execution

When the train gets information from execution balise, or the train estimation front end

passes the advance notice execution position, and the train running speed is lower than the CTCS-2 level allowable speed, the train begins to perform the level transform. If the level transform fails, the brake command is output to stop the train. After the transform completed, the RBC sends a termination communication session to the ATP when the tail of the train passes the transition boundary. If the ATP does not receive the disconnect command, the ATP disconnects the RBC after sending the location report twice.

IV. SCADE-BASED LEVEL TRANSFORM MODELING

According to the transform process of CTCS-3 to CTCS-2, the level transform model established in this paper is shown in figure 4. The train receives the level transform notice command (HandleLTA) as the initial state, and judges the relationship between the train speed (TrainSpeed) and the allowable speed of the CTCS-2 level (C2PermitSpeed). If information is got from execution balise (LTOdetected=true) or the specified position is reached (LTOpassed =true), and the front end of the train reaches the boundary (TransBoundPassed=true), the level transform is started. If the transform fails (TransSucceed=false), or the direction of the train changes (DirecChanged = true) or remains unchanged without entering the notice area (DirecChanged = false and LTApassed=false), the CTCS-3 level operation will be maintained. If the transition is successful (TransSucceed=true), when the full length of the train passes the transition boundary (WholeBoundPassed=true), the ATP ends the communication session with the RBC after receiving the RBC disconnect command (RBCdisconnected=true). If the RBC disconnect command is not received, the communication session ends with the RBC after the location report is sent. At this point, the CTCS-3 level transition to the CTCS-2 level is complete.

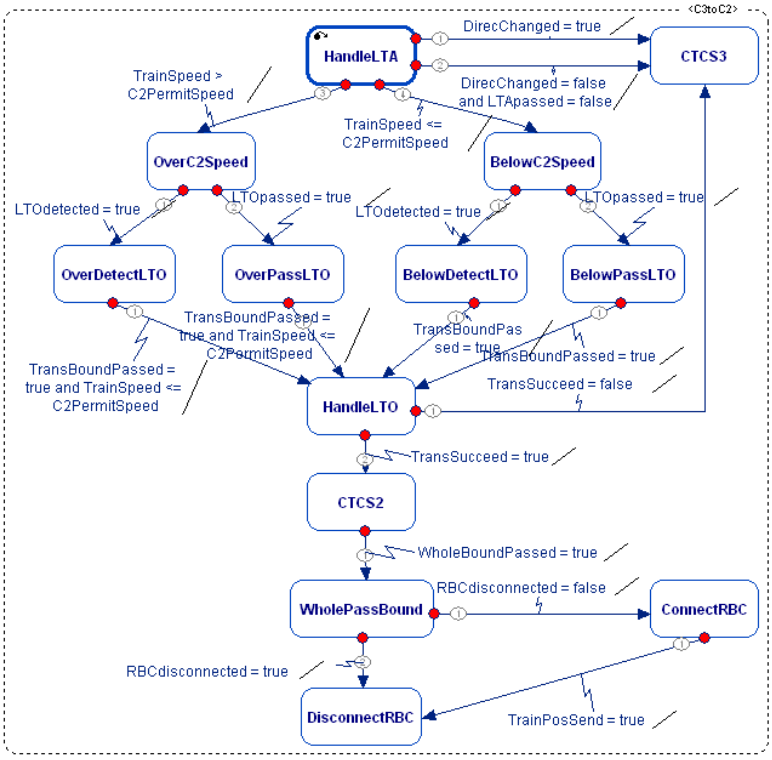
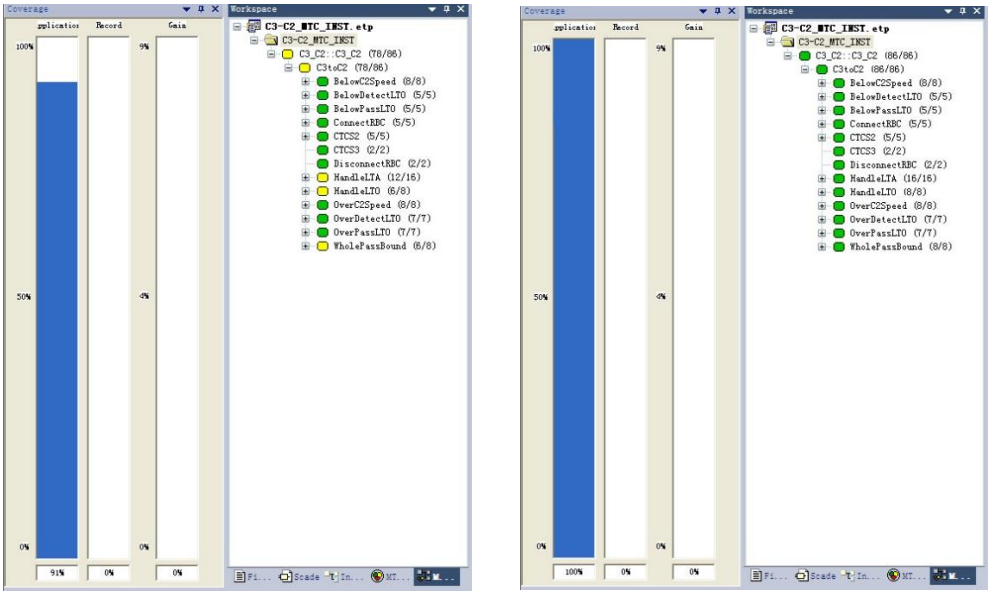


Figure 4. *Level Transform Model*

V. SCADE-BASED LEVEL CONVERSION TEST CASE GENERATION

This paper uses the model coverage analysis tool MTC to automatically generate test cases. By setting the input variable of each period and simulating the model, the test case coverage is dynamically observed. The model coverage increases each time a new model variable is entered. At the same time, the MTC will give the coverage information of each node. For the nodes that are not covered, the MTC will prompt the relevant variable information. As shown in figure 5-a, the coverage rate of the model is 91%. The possible reasons are that the coverage effect of nodes such as WholePassBound, HandleLTA, HandleLTO is not good. By adjusting the relative strain of each node in the model, such as Whole Bound Passed, TransSucceed, TransBound Passed, and so on, the complete coverage of the nodes can be improved, so as to achieve 100% coverage of the model, as shown in figure 5-b. A total of 14 test cases were obtained by this method, one of which was shown in Figure 6. The test case lasts for 6 cycles, and the elapsed state is: HandleLTA→BelowC2Speed→BelowDetectLTO→HandleLTO→CTCS2→WholePassBound→DisconnectRBC. The test case shows that after the train receives the level transform notice, the running speed is 40km/h, which is lower than the CTCS-2 level allowed speed. After the information is got from execution balise and the front end of the train passes the transition boundary, the level transform is successful. When the train gets the RBC disconnected command, it disconnects from the RBC successfuly.



a. *Not Fully Covered* ***b.*** *Fully Covered*

Figure 5. *Coverage Verification*

```
SSM::set C3_C2::C3_C2/TrainSpeed 40
SSM::set C3_C2::C3_C2/TrainPosSend false
SSM::set C3_C2::C3_C2/DirecChanged false
SSM::set C3_C2::C3_C2/RBCdisconnected true
SSM::set C3_C2::C3_C2/LTApassed true
SSM::set C3_C2::C3_C2/WholeBoundPassed true
SSM::set C3_C2::C3_C2/TransSucceed true
SSM::set C3_C2::C3_C2/TransBoundPassed true
SSM::set C3_C2::C3_C2/LTOdetected true
SSM::set C3_C2::C3_C2/LTOpassed false
SSM::cycle 6
```

Figure 6. *Partial Test Case Diagram*

The autonomous ATP level transform test case is manually written by the tester and there is no specific generation method. By modeling the CTCS-3 level to CTCS-2 level transform with SCADE, an effective test environment can be analyzed, which can more easily represent complex level transform function relationships, and is easy for professional and non-professionals to understand the level transform function. It contributes to the development and design of ATP software functions. At the same time, the MTC generation test case method proposed in this paper can directly generate effective test cases covering all scenarios, and also makes testers clear their test objectives and test tasks, thereby reducing the influence of human factors on test adequacy and improving the test efficiency of train control system.

VI. CONCLUSIONS

This paper proposes a test case generation method based on SCADE. This method uses SCADE to accurately model complex systems, observe node coverage information through MTC, and modify the model input variable simulation to dynamically adjust the coverage of test cases, thus helping testers to obtain optimal test cases.

ACKNOWLEDGEMENT

This research was supported by National Natural Science Foundation of China (Grant No. 61703349), Key Research Projects of China Railway Corporation (Grant No. N2018G062, K2018G011), and Fundamental Research Funds for the Central Universities (Grant No. 2682017CX101).

References

[1]. *Z.J. Yang,* "Research on Autonomy of Key Equipment in CTCS-3 Train Control System," China Railway, no.7, pp.1-6, 2018.

[2]. *K. Feng, J.F. Cheng, and L. Yue,* "Study of Standards & Specifications of CTCS-3 Onboard Equipment," China Railway, no.9, pp.1-4, 2018.

[3]. *S.H. Li, J. Wang, Y.C. Qi,* "Model-Based Methods for Real Time System Testing," COMPUTER ENGINEERING & SCIENCE, vol.28, no.4, pp.119-123, 2006.

[4]. *J. Yang, J. Wang, H.W Chen,* "A review of Model-Based for Software Testing," COMPUTER SCIENCE, vol.31, no.2, pp.184-187, 2004.

[5]. *S.Z. Chen, R.W. Chen, Y. Li,* "Method of SCADE-based safety software development," Railway Computer Application, vol.3, no.24, pp.14-18, 2015.

[6]. *J. Dong, S.H. Dai,* "Modeling and formal analysis of level transition based on colored Petri nets," Computer Engineering and Applications, vol.2, no.54, pp.208-213, 2018.

[7]. *A.H. Zhu, L.M. Song,* "Modeling and formal analysis of level transition in train control system based on UML and CPN," Application Research of Computers, vol.1, no.36, pp.140-143+162, 2019.

[8]. *R.W. Kang, J.F. Wang, J.D. Lu,* "UPPAAL-based modeling and verification of level transition process of high-speed railway train control system," Journal of Beijing Jiaotong University, vol.6, no.36, pp.63-67+73, 2012.