International Journal of Control and Automation
Vol.8, No.10 (2015), pp.171-178
http://dx.doi.org/10.14257/ijca.2015.8.10.17

Model-Based Test Case Generation for Function Testing of CTCS-3 Onboard Subsystem

Jidong Lv, Pengcheng Ren, Chen Lei, Kaicheng Li and Tao Tang

National Engineering Research Center of Rail Transportation Operation and Control System, Beijing Jiaotong University, China
Beijing, China
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, China
Beijing, China
School Of Electronics, Electrical and Computer Engineering,
The University of Birmingham,
UK
Birmingham, Uk

Abstract

The CTCS-3(Chinese Train Control System level 3) is an automatic system which is an integrated of advanced control technology, advanced communication technology, advanced computer technology and railway signal technology. It plays an important role in assuring safety and improving efficiency in railway. As a core subsystem in CTCS-3, the onboard subsystem is a typical safety-critical system, in which any fault can lead to huge human injury or wealth losing. Function testing method which is mainly focus on the conformance relation between the specification and the System Under Test (SUT) has been widely used in testing onboard subsystem in the past few years. However, most of the test cases are manually generated which can't be reused and leads to repeated works when the specification is changed.

To improve the testing efficiency and quality, model-based testing method is introduced. We use a tool chain to generate test case automatically based on Timed Automata theory and then apply in function testing of onboard subsystem. Firstly, we establish the scenario-tree by analyzing the specification of onboard subsystem and Vital Computer and Environment (V-E) timed automata network model of mode transition using tool Uppaal. Then, according to the relation between train modes and operational scenarios, test cases are automatically generated by tool CoVer with definition coverage criteria based on the Observer Automata theory. Finally, a selection algorithm is given to choose a complete test sequence from the test cases.

Different test suitss of onboard subsystem are acquired and compared with different coverage criteria and coverage items. A complete model transition function test suits are derived which is proven very useful for testing the onboard subsystem.

Keywords: *Model-Based Testing, Test Suite, On Board, Uppaal, Cover*

1. Introduction

As a core system in CTCS-3, the onboard system is a typical safety-critical system, responsible for the implementation of speed protection and safe distance between trains, in which any fault can lead to huge human injury or wealth losing [1]. It is very important to ensure the correctness of functions in the onboard system. Function testing method

ISSN: 2005-4297 IJCA
Copyright ⓒ 2015 SERSC

which is mainly focus on the conformance relation between the specification and the SUT has been widely used in testing the onboard system in the past few years.

Test case is the basis of testing and how to automatically generate test cases that satisfy the system specifications completely is the key issue of the onboard system test. Most of the traditional test cases are manually generated which can't be reused and leads to repeat works when the specification is changed [2]. With the widespread of formal method, a lot of automatic test case generated technologies are introduced, and scenario-based testing has been gradually used. In scenario-Based testing, scenario techniques[3,4] are applied to the test case expression, and the conversion tool will automatically generates test cases that can be run on a certain test platform and achieve the purpose of the test. Scenario-based test cases make the system easier to understand, and are widely used in the functional testing of the system. Wang Shuai changed Unified Modeling Language (UML) sequence diagram to automata network model with scene technology, and established a suitable formal modeling method for train control system [5]. Huang long achieved finite state machine from UML sequence diagram to specify the system behavior and got the scenario-based formal testing model [6]. Yin Yongfeng applied scenario based test case generation method to embedded software and optimized the embedded system test automation [7]. Pan Jianyong proposed a method to optimize scenarios, and applied it to the safety test of TSRS with XML language [8].

In this paper, to improve the testing efficiency and quality of the onboard system, a scenario based test generation algorithm is given in order to generate a complete test suits of the function of the onboard system. We use a tool chain (Uppaal & CoVer) to generate test cases automatically based on scenario technique. Then, the choosen algorithm is developed to achieve the test sequence covering all modes of onboard system. The complete model transition function test suits are derived which is proven very useful for testing the Onboard system.

2. Modeling of Onboard Subsystem

2.1. Scenario Tree Model

Operating scenarios described the behavior and the expectative running modes of CTCS-3 in the view of train [9]. In order to get the running mode test cases for onboard system, analysis and management of the operating scenarios in CTCS-3 specification are needed. The modeling process is shown in Figure 1, all scenes are classified in scenario tree: system level, objective level and property level.

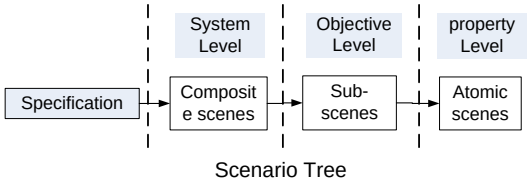


Figure 1. Scenario Technology

1) System level represents a function packet, reflecting the abstractive function set of the system. It consists of several operating scenarios, called composite scene;
2) Objective level represents the specific function item and the related condition and operation, called sub-scene;
3) Property level represents the minimum function of the system and can be tested independently, called atomic scene.

 Copyright ⓒ 2015 SERSC

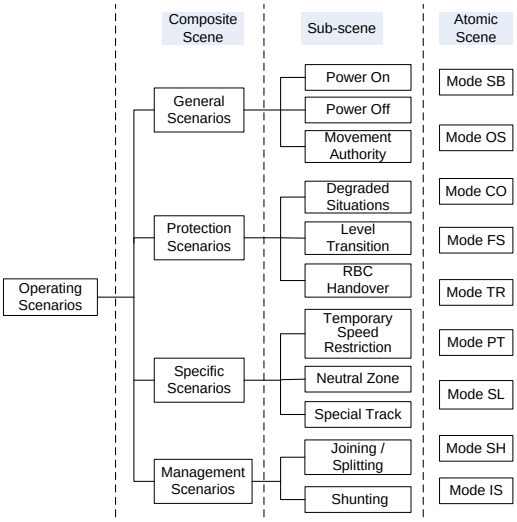


Figure 2. Scenario Tree Model of Onboard Subsystem

According to above modeling technology, we describe onboard system function using hierarchy scenario tree in Figure 2. In system level, the function scenarios are classified as general scenarios, protection scenarios, specific scenarios and management scenarios. In objective level, we match composite scene of CTCS-3 with the operating scenarios, including Power On, Power Off, Level Transition, Movement Authority, RBC Handover, Pass Splitting Phase, Joining /Splitting, Temporary Speed Restriction, Degraded Situation, Shunting and Special Track. The train operating scenarios are expressed by the train operating modes and their transitions, which are used as the atomic scenes. In CTCS-3, nine operating modes are Isolation mode (IS), Sleeping mode (SL), Stand-By mode (SB), Shunting mode (SH), Full Supervision mode (FS), On Sight mode (OS), Call On mode (CO), Trip mode(TR) and Post Trip mode(PT), and their transition conditions between them is described in [10].

Table 1. Relationship between Operating Scenarios and Modes

Operating Scenarios	Mode and Transtions
Power On	(service end) SB-CO / SB-FS (non-service end) SB-SL
Power Off	SB-SL
Movement Authority	SB-FS/CO PT-FS/CO OS-FS/CO
Level Transition /RBC Handover	FS / CO
Pass Neutral Zone / Temporary Speed Restriction	FS / CO / OS
Joining / Splitting	(Joining) FS-SB-SH-SB (Splitting) FS-SB
Degraded Situation	FS /SB/CO/PT - OS
Shunting	SH
Special Track	IS

The relation between sub-scene and atomic scene of scenario tree is depicted in Table 1. Each scenario contains one or more running modes, so the mode transition can be described with scenario. For example, mode SB to mode CO can be achieved in the scenario, guiding track departure instruction after power on.

3. Network Timed Automata Model

The on-board equipment is a computer-based subsystem that supervises the movement of the train to which it belongs, on basis of information exchanged with the trackside sub-

Copyright ⓒ 2015 SERSC

system [10]. The interoperability requirements for the on-board equipment are related to the functionality and the data exchange data exchange between the on-board sub-system and the driver, the train, the specific transmission modules. The described structure is depicted in Figure 3.

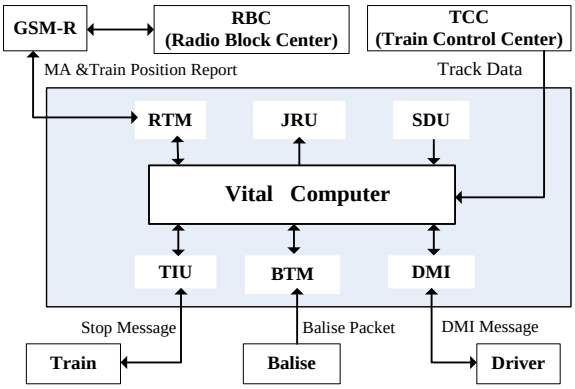


Figure 3. Architectural Scheme of Onboard Subsystem

As a method to describe complex discrete event system, automotive is suitable for modeling and analysis of train control system. Onboard equipments consist of vital computer, train, RBC, balise and driver. Based on the structural analysis of Onboard subsystem, the entire system network automaton model is divided as kernel part and environment, VC ||Environment(Train, Driver, RBC, Balise), named V-E model.

Based on the atomic scenes, the network automata model of the Onboard subsystem is established with Uppaal tool [11], according to the mode transition table, symbolic description and conversion conditions in chapter 4.4 of specification. VC automata, Train automata, Balise automata, RBC automata and Driver automata are parallel and their respective modeling variables are defined. For example in VC automata, we define T_{VC} =<S , S^0,A,X,I,E >, in which S={SB , SH ,FS ,OS ,CO , SL ,TR ,PT , IS}, the location and edge is defined as the following symbols:

1）Location: <l，σ>, l is the currant location and σ is related variables of location l;

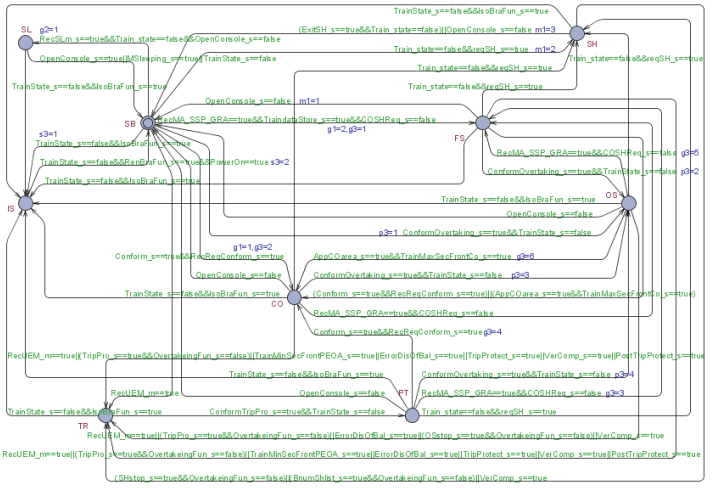
2） edge: $e:< l, \sigma > \xrightarrow{a} < l', \sigma' >$, location l will change to l’ and variable set σ to σ’ with condition a.

Table 2. Scenario Variables

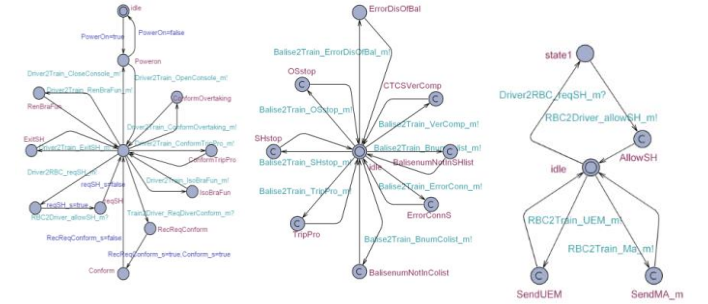
Scenario	Symbol	Operating Scenarios
General Scenarios	G={g1，g2，g3}	{Power On,Power Off,Movement Authority}
Protection Scenarios	P={p1，p2，p3}	{Level Transition,RBC Handover , Degraded Situation}
Specific Scenarios	S={s1，s2，s3}	{Temporary Speed Restriction, Pass Neutral Zone, Special Track}
Management Scenarios	M={m1，m2}	{Joining / Splitting, Shunting}

For the sub-scene level, operating scenarios are modeled in VC automata according to the procedure described in chapter 3 of specification [10]. The scenarios variables are defined with set G, P, S and M, depicted in Table 2.

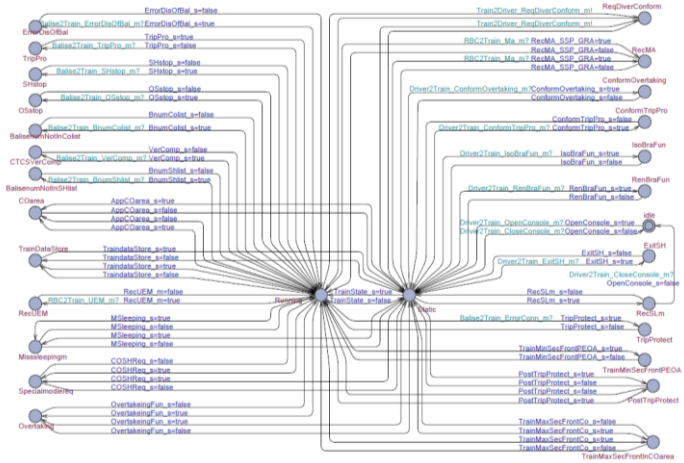
 Copyright ⓒ 2015 SERSC



a) VC Automata



b) Driver Automata c) Balise Automata d) RBC Automata



e) Train Automata

Figure 5. V-E Timed Automata Model

The V-E automata model is shown in Figure 5, a) is VC automata and others are the four parts of environment automata, providing the mode transition conditions to VC automata. The information between On-board system and Train is train braking information and train state information, between On-board system and RBC is MA message and train location, between onboard system and balise is train location and Hazard protection, between On-board system and Driver is the conform and choice of operation.

Copyright ⓒ 2015 SERSC

We used BNF language in Uppaal to check the logic of the model itself and it came out that there was no deadlock in the V-E model and each mode is reachable.

4. Test Sequence of Onboard Subsystem

4.1. Test Cases Generation with Cover

Observer can be used to specify coverage criteria for test generation and monitor the location and edge in test trace. Whenever the coverage item has been covered, the observer location is called “accepting location”. The accepting set records all the test items matching the coverage criteria described by the observer and it will guide test cases generation as a configuration file in CoVer [12] (developed by Andrers Hessel in Uppsala University). Thinking about feature of V-E automata network model and Observer Automata theory [13], we proposed test coverage criteria in the fallowing Figure 7.

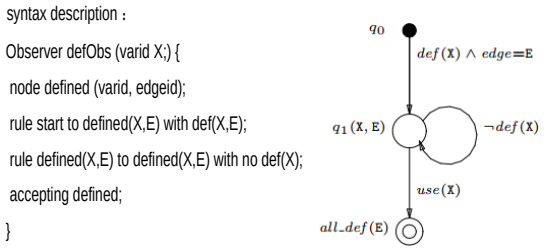


Figure 7. All-Definitions Coverage Observer

In the VC automata, G, P, S, M are used to represent the scenarios of onboard subsystem, so X in the syntax above is{ g1,g2,g3,p1,p2,p3,s1,s2,s3,m1,m2}. Then, the test suits were automatically generated by CoVer. See in Table 3.

Table 3. Test Suits of Scenarios

Test suits		
Variable	Test Item	Transition Trace
g1	defined<varid g1, edgeid VC. SB_to_CO>	VC.SB ->VC.CO ->VC.SB ->VC.SL
	defined<varid g1, edgeid VC.SB_to_FS>	VC.SB ->VC.FS ->VC.SB ->VC.SL
g2	defined<varid g2, edgeid VC.SB_to_SL>	VC.SB ->VC.SL ->VC.SB ->VC.CO ->VC.TR ->VC.PT
g3 p1,p2 s1,s2	defined<varid g3, edgeid VC. OS_to_FS>	VC.SB ->VC.OS ->VC.FS ->VC.TR
	defined<varid g3, edgeid VC.SB_to_FS>	VC.SB ->VC.FS ->VC.SB ->VC.SL
	defined<varid g3, edgeid VC. PT_to_CO>	VC.SB ->VC.TR ->VC.PT ->VC.CO ->VC.SB ->VC.SL
	defined<varid g3, edgeid VC.SB_to_CO> defined<varid g3, edgeid VC.PT_to_FS>	VC.SB ->VC.CO ->VC.TR ->VC.PT ->VC.FS ->VC.SB ->VC.SL
p3	defined<varid p3, edgeid VC.SB_to_OS>	VC.SB ->VC.OS ->VC.SB ->VC.SL
	defined<varid p3, edgeid VC.CO_to_OS>	VC.SB ->VC.CO ->VC.OS ->VC.SB ->VC.CO
	defined<varid p3, edgeid VC.FS_to_OS>	VC.SB ->VC.FS ->VC.OS ->VC.SB ->VC.SL
	defined<varid p3, edgeid VC.PT_to_OS>	VC.SB ->VC.TR ->VC.PT ->VC.OS ->VC.TR ->VC.PT
s3	defined<varid s3, edgeid VC. SB_to_IS> defined<varid s3, edgeid VC.IS_to_ SB>	VC.SB ->VC.IS ->VC.SB ->VC.CO ->VC.TR ->VC.PT
m1 m2	defined<varid m1, edgeid VC.FS_to_ SB>	VC.SB ->VC.FS ->VC.SB ->VC.CO ->VC.TR ->VC.PT
	defined<varid m1, edgeid VC.SB_to_SH> defined<varid m1, edgeid VC.SH_to _SB>	VC.SB ->VC.SH ->VC.SB ->VC.CO ->VC.TR ->VC.PT

 Copyright ⓒ 2015 SERSC

Here we need to explain that p1 (Level Transition), p2(RBC Handover), s2(Pass Neutral Zone)and s1(Temporary Speed Restriction) can be executed in mode FS, so their test items can be found in g3(Movement Authority) test items.

5. Algorithm for Test Sequence Generation

The scenario trace is based on the operating scenes and described as test suits, which shows the mode transition. Effective test sequence of onboard subsystem can be made up by selecting mode coverage test cases from the test suits in CoVer .Thus, the problem of finding test sequence is become to find the following trace that covers in the accepting location set Q of observer.

$$tr = << l_0, \sigma_0 > \| \{q_0\} > \xrightarrow{a} \ldots \xrightarrow{a'} << l, \sigma > \| Q >$$

A algorithm was designed to get test sequence covering modes entirely:

```
Construct an Observer model B, Q;
Construct an trace TR;
i = 0; j = 0;
while(i < ScenarioNum) do
  Select test suit Ai : includes most modes;
  Max = TotalTrace(Ai);
  REPEAT
    if exist one mode in Ai.trj : qx ∉ Q then
       Add TestCase_Mode (Ai.trj.q, Q);
       Store TestCase_Edge (Ai.trj, TR);
    j = j +1;
    Find_nextTrace(Ai.trj);
  UNTIL  j = Max;
  i = i +1; j = 0;
RETURN  TR and Q;
```

The programmed algorithm is described as following: for any certain scenario test case set A, get its trace number MaxTrace, assume the accepting location set of observer is Q ,and our target sequence is TR. Define the variable i=0, j=0.

1) Add $A_i.tr_j$ to TR and tr.mode to Q if there exist new mode in the test trace $A_i.tr_j$;
2) Make j=j +1, do Find_nextTrace(tr_j) and find next trace of A, ;
3) Execute the step 1 and 2 until j= MaxTrace;
4) Make i=i+1 and go on the same work for next scenario;
5) Return target sequence TR and accepting set Q.

Then, the target sequence was generated from the test suits with the algorithm above, see in Table 4.

Table 4. Test Sequence of All Mode Coverage

Test Squence	
Test Case	Test Item
#1	edgeN<edgeid VC.SB_to_CO> edgeN<edgeid VC.CO_to_TR> edgeN<edgeid VC.TR_to_PT> edgeN<edgeid VC.PT_to_FS> edgeN<edgeid VC.FS_to_SB> edgeN<edgeid VC. SB _to_SL>
#2	edgeN<edgeid VC.SB_to_SH> edgeN<edgeid VC.SH_to_ SB> edgeN<edgeid VC.SB_to_CO> edgeN<edgeid VC.CO_to_TR> edgeN<edgeid VC.TR_to_PT>
#3	edgeN<edgeid VC.SB_to_IS> edgeN<edgeid VC.IS_to_SB> edgeN<edgeid VC.SB_to_CO> edgeN<edgeid VC.CO_to_TR> edgeN<edgeid VC.TR_to_PT>
#4	edgeN<edgeid VC.SB_to_FS> edgeN<edgeid VC.FS _to_OS> edgeN<edgeid VC.OS_to_SB> edgeN<edgeid VC.SB_to_SL>

Copyright ⓒ 2015 SERSC

6. Conclusion

As most of the test cases are manually generated which can't be reused and leads to repeated works when the specification is changed, Model-based testing method is introduced. We use a tool chain to generate test case automatically based on scenario technique and timed automata theory to improve the testing efficiency and quality, and then apply in function testing of onboard system. V-E Model is built to interpret the structure and behavior of the onboard system specification, and it is realized with tool Uppaal. Then, coverage criteria are used in tool of CoVer to generate test case automatically, and choosing algorithm is developed to achieve the test sequence covering all modes of onboard system. Finally a complete mode transition function test sequence is derived which is proven very useful for testing the onboard system.

Acknowledgments

The authors would like to thank Lei Yuan and Yu Liu from SKLRTCS, Beijing Jiaotong University for their useful com-ments on case study. The research of the work reported here were supported by the National Basic Research Program of China (973 Program): Research on the construction and quality ensurence of safety-critical software system (2014CB340703); the National Natural Science Foundation of China: Model-based conformance testing on function of train control system (61304185)" ; the Fundamental Research Funds for the Central Universities" : "Research on formal model conformance testing and its application on safety function testing of High-speed railway"(2014JBM022).

References

[1] LV Jidong and Tang tao, "Modeling and Verification of Time Constraints of Operation Scenarios of High-speed Train Control System[J]", JOURNAL OF THE CHINA RAILWAY SOCIETY. 2011.
[2] B. Beizer, "Black-Box Testing Technique for Functional Testing of Software and Systems", Wiley,New York, USA9,1995.
[3] Y. Guanghua, Q. Xuan and S. Yansheng, "Scene mode based test cases design for the embedded software [J]. Computer Engineering, 2010, 36（15）.
[4] M. Lettrari and J. Klose. "Scenario-Based Monitoring and Testing of Real-Time UML Models[C]", ,4th International Conference Toronto, Canada, October 1–5, 2001 Proceedings. Springer-Verlag Berlin Heidelberg,2001:317-328.
[5] W. Shuai, J. Yingdong and Y. Shiyuan, "Scenario-based modeling method for CTCS-3 train control system [J]", JOURNAL OF THE CHINA RAILWAY SOCIETY, 2011,33（9）.
[6] H. Long and M. Huaikou, "Scenario-based formal test modeling method [J]", Shanghai University, 2011,17（5）.
[7] Y. Yongfeng, L. Bin, J. Tongmin, "Scene Based test cases generation method for embedded software [J]",Computer Engineering and design, 2008,29（16）.
[8] P. Jianyong and C. Bangxing, "Scenario-based test case design study", Communication technologies, 2011, 44（12）.
[9] S. Zhang, "The general technical Programme of CTCS-3 level train control system in railways for passengers [M]", China Railway Publishing House,2008
[10] Department of railway ministry science and technology. CTCS-3 System Requirement Specification [M]. Beijing: China Railway Publishing House, 2009.
[11] G. Behrmann, K.G. Larsen, O. Moller, A. Divid, P. Pettersson and W. Yi, "UPPAAL - present and future//In the Proc of the 40th IEEE Conference on Decision and Control Orlando", Florida USA, December 2001, pp. 2881-2886.
[12] A. HESSEL and P. PETTERSSON, "COVER—A Real-time Test Case Generation Tool[Z]", Accepted for the 3rd Workshop on Model-Based Testing 2007 (MBT07).
[13] J.Blom, A. HESSEL and P. PETTERSSON, "Specifying and Generating Test Cases Using Observer Automata[C] GABOWSKI J", NIELSEN B. Proceedings 4th International Workshop on Formal Approaches to Testing of Software 2004 (FATES,04), Volume 3395 of LNCS. Springer-Verlag, 2005: 125-139.

 Copyright ⓒ 2015 SERSC