

· 新能源汽车与低碳运输 ·

基于模型的车载设备测试用例自动生成研究

易承龙, 李开成, 周晶晶

(北京交通大学轨道交通运行控制系统国家工程研究中心, 北京 100044)

摘 要:针对人工生成测试序列的不足, 提出基于模型的车载设备测试用例自动生成方法。首先按照系统需求规范, 在 UPPAAL 环境下运用时间自动机对车载设备进行建模及验证, 然后将建立的模型导入到基于覆盖度算法的模型辅助工具 Cover 中自动生成测试用例, 最后分析了自动生成的测试用例的正确性。

关键词:车载设备; 测试; UPPAAL; Cover; 测试用例; 时间自动机

中图分类号:U284.91 **文献标志码:**A **文章编号:**1673-159X(2014)02-0058-05

doi:10.3969/j.issn.1673-159X.2014.02.013

Research on Automatic Generation of Test Cases for on-board Equipment CTCS-3Based on Model

YI Cheng-long, LI Kai-cheng, ZHOU Jing-jing

(National Engineering Research Center of Rail Transportation Operation and Control System, Beijing Jiaotong University, Beijing 100044 China)

Abstract: Aimed at the shortcomings of the artificial generating test sequences, In this paper, the authors put forward a method based on model. First according to the system requirements specification, under the environment of UPPAAL, the timed automaton is used to model the on-board equipment and check the model, then the model is put into Cover which is a tool based on coverage algorithm to automatically generate test cases, finally, the correctness of the automatic generation of test cases is analyzed.

Keywords: on-board equipment; test; UPPAAL; Cover; test case; timed automaton

2004年,铁道部提出“四横四纵”铁路快速客运专线网建设规划,从此启动了中国高速铁路发展计划。随着铁路事业的快速发展,列控系统的安全性也成为了人们关注的焦点^[1]。作为列控系统核心的车载设备在正式使用前必须进行全面测试,测试的基础是测试用例的生成。目前,车载设备测试用例是根据专家经验通过启发的方式得到,这种方式既耗时,又不能保证测试用例的覆盖度^[2];因此,有必要在保证测试覆盖度的前提下将测试用例的生成过程自动化。

1 列控系统车载设备测试现状

现阶段,CTCS-3级车载设备的测试是通过系统既有的外部接口进行功能测试,采用的是黑盒测试及主动测试方法^[3]。功能测试由系统功能需求规范提取出功能特征,依据功能特征编写测试用例,然后由测试用例串联成测试序列^[3-5]。目前,由功能特征设计测试用例的过程主要由手工完成,测试的效率低、耗时长,并且随着测试规范和测试目的的改变,测试工作量增大。测试用例编写人员需要对系统深入了解,并且具备丰富的测试经验,即使

收稿日期:2013-03-20

基金项目:国家科技部“863”项目(2012AA112801)。

作者简介:易承龙(1988-),男,硕士研究生,主要研究方向为高速列车运行控制系统的仿真及测试。

1) 功能要求。

$A[]$ not deadlock

系统没有死锁;

$E < > ((VC.VCWaitM3) \text{ and } (RBC.RBCWaitM132))$

RBC 能定时接受车载发送的 MA 请求并向车载发送最新 MA;

$E < > ((VC.VCWaitM24) \text{ and } (RBC.RBCWaitM136))$

车载设备能定时向 RBC 发送列车实时位置信息并收到 RBC 回复。

2) 性能要求。

$A[] VC.VCWaitM3 \text{ imply } T_MaReq > MAReqPer - iod$

车载设备周期向 RBC 发送请求 MA; $A[] VC.VCWaitM24 \text{ imply } T_PositionReport >$

PosRepPeriod

车载设备周期向 RBC 发送位置报告信息;

$A[] VC.EditonJudgeStu \text{ imply } ConnectNum < = TryToConNo$

车载设备与 RBC 建立安全连接尝试呼叫的次数小于 3 次;

$A[] VC.EditonJudgeStu \text{ imply } T_Con < = T_NVCONTACT$

车载设备与 RBC 建立安全连接时间不超过 60 s。

在 UPPAAL 软件验证器中执行上述验证语句,结果显示上面所描述的性质都能通过,表明系统满足这些性质,模型是安全正确的,这为后面测试用例自动生成提供了正确的前提。

4 车载设备测试用例的自动生成

本文首先介绍了基于模型测试的原理和过程,然后引入基于覆盖度算法的辅助软件 Cover,自动生成了车载设备的测试用例。

4.1 基于模型的测试原理

首先叙述一个事实:基于模型的测试序列自动生成方法应用的前提条件是假设系统是 TITOS 系统,也就是说对于每一个测试用例的输入,系统都有一个对应于系统详细设计中所描述的明确输出,车载设备满足这个条件。假设系统 A 是实时系统,它由 2 个部分组成,分别是系统模型 S 和系统所在的环境 E。时间自动机在运转时,组成系统的各个部件在时钟约束的条件下从一个状态到达另一个

状态形成一条序列,而特征轨迹就是序列的集合。利用 UPPAAL 对可达性问题分析产生的特征轨迹可用以下数学公式表达:

$$(s_0, e_0) \xrightarrow{r_0} (s_1, e_1) \xrightarrow{r_1} (s_2, e_2) \xrightarrow{r_2} \dots (s_n, e_n)$$

其中: s_i 和 e_i 分别是系统模型 S 和运行环境模型 E 的状态; r_i 是时间延迟或者其他同步动作。系统延迟和其他同步动作组成了测试用例中的事件因子,因此从特征轨迹到测试用例的转换,最简单的方法就是通过将特征轨迹映射到系统环境模型 E 中,叠加相关联的系统延迟,去掉不可见的转换过程,最后,将要在真实环境中执行的测试用例引入判决。其判决过程可以解释如下:时间自动机中某个状态用一个 Bool 类型变量标识,可以标识 Pass 和 Fail 状态,如果测试序列不能执行到这个状态,那么他将被标识为 Fail 状态,如果能执行经过这个状态,他将被标识为 Pass 状态。

4.2 车载设备测试用例的自动生成

4.2.1 Cover 工具配置

在用 Cover 自动生成测试序列之前,必须对 Cover 进行相应的配置,相关配置参数如表 1 所示。

表 1 Cover 工具配置参数

参数名称	参数作用
-G	使用通用的覆盖标准算法
-M n	n 用于标记测试案例最大的覆盖范围
-K	用于打印生成文件的参数
-t	记录测试序列运行轨迹
-f	保存测试序列运行轨迹文件的目录

Cover 工具需要输入 Model.xml 文件、Observer.obs 文件和 Property.q 文件。

Model.xml 文件即上一节使用 UPPAAL 软件建模所生成的文件,需要拷贝到 Cover 工具所在的根目录下。

Observer.obs 文件用于描述覆盖标准算法。其使用非常灵活,具体语法结构形式如下所示,以下所描述的为全位置覆盖标准。

```
observerObserver(procId P; )
```

```
{
    node edgeNode(edgeId);
    rulestartto edgeNode(E) with E := edge(P);
    accepting edgeNode;
}
```

Property.q 文件用于配置 Cover 工具选择生成

哪个时间自动机模型的测试例,其语法结构形式如下:

CoveredgeObs({P1,P2})

在这里,时间自动机过程集为{P1,P2},也就是说 P1 和 P2 会分别代入 Observer 文件中的 procid P 进行运算。

4.2.2 测试用例的自动生成

根据上一节描述的配置方法配置 Cover 工具,并将 XML 格式的注册与启动场景的 UPPAAL 模型和使用全位置覆盖标准算法的 Observer.obs 文件拷贝到 Cover 程序所在的文件夹的根目录,运行 Cover 工具。将生成的测试案例用例保存在 abc.tr 文件中,运行结果表明共生成了 7 个测试案例。下面以 Trace#1 为例,介绍所生成的抽象语言描述的具体含义。Trace#1 的具体描述如下:

• Trace #1:

edgeN < edgeid VC. EVCIInActivity_to_StandBy >

edgeN < edgeidVC. StandBy_to_ReadyToConRBC >

edgeN < edgeidVC. ReadyToConRBC_to_WatiRBCM32 >

edgeN < edgeidVC. WatiRBCM32_to_TimeOutJudgeStu >

edgeN < edgeidVC. TimeOutJudgeStu_to_EditonJudgeSt >

edgeN < edgeidVC. EditonJudgeStu_to_ConWithRBC >

edgeN < edgeidVC. ConWithRBC_to_WaitParameter >

可以将以上所生成的抽象状态转移语言转换具体的状态转移图,这样所生成的测试用例的消息序列为:“初始状态”—“待机状态”—“准备建立连接状态”—“等待消息 32 状态”—“判断连接是否超时状态”—“判断系统版本状态”—“建立连接状态”—“等待配置参数状态”—“等待输入列车数据状态”—“等待列车数据确认状态”—“等待发车状态”—“正常工作状态”—“等待行车许可状态”—“发送列车位置报告状态”。

通过对比《CTCS-3 级列控系统测试案例》中

功能特征 47 的测试案 1 的基本信息和测试步骤,可以发现,其所描述的事件与 Cover 工具所生产的 Trace#1 相吻合。由上面的案例可知,利用时间自动机建模不仅可以验证模型的可达性,还可以通过搜索路径生成覆盖度完整的测试用例,并且可以通过配置 Cover 工具搜索算法改变测试用例的路径选择。

5 结束语

传统的车载系统功能测试用例生成主要依靠专家经验手工编制完成,测试的主观性强、效率低、耗时长,并且随着测试规范和测试目的的改变,测试工作量大,而且测试的覆盖度无法得到满足。本文提出了一种基于模型的车载系统安全测试用例自动生成方法,使用 UPPAAL 软件对列车注册与启动运营场景进行建模,并从功能和性能 2 方面对所建立的模型进行了验证,最后运用基于覆盖度的工具 Cover 自动生成车载设备测试用例,这为复杂时变系统测试用例的自动生成提出了一种新的思路。

参考文献

- [1] 唐涛. 高速铁路列车运行控制系统车载设备安全性设计[J]. 北方交通大学学报,1999;23(5):83-87.
- [2] 王超琦. CTCS-3 级列控系统车载设备测试序列的生成方法及工具研究[D]. 北京:北京交通大学,2010;5-27.
- [3] Beizer B. Black-Box Testing: Techniques for Functional Testing of Software and Systems[M]. New York:John Wiley & Sons;1995:160-165.
- [4] 王倩倩. CTCS-3 级列控系统测试案例优化生成方法研究[D]. 北京:北京交通大学,2010;16-17.
- [5] 章慧. CTCS-3 级列控系统车载设备测试方法研究[D]. 北京:北京交通大学,2007.
- [6] 范素娟. 基于时间自动机模型的测试用例生成方法研究[D]. 郑州:郑州大学,2010.
- [7] Alur R, Dill D L A. Theory of Timed Automata[J]. Theoretical Computer Science,1994,126:183-235.
- [8] 范素娟,庄雷. 基于时间自动机模型的测试用例生成方法[J]. 计算机工程与设计,2010(12):2765-2768.
- [9] 张曙光. CTCS-3 级列控系统总体技术方案[M]. 北京:中国铁道出版社,2008;6-8.

(编校:夏书林)