

# 基于接口自动机的列车通信设备测试用例生成

郁 莲<sup>1</sup>,李军龙<sup>1</sup>,白晓颖<sup>2</sup>,刘 鑫<sup>1</sup>,吴中海<sup>1</sup>

(1. 北京大学 软件与微电子学院,北京 102600;

2. 清华大学 计算机学院,北京 100084)

**摘 要:**列车通信设备是一种嵌入式系统。有效地测试这类设备是列车运行系统质量保证的重要手段。嵌入式系统测试的重要任务之一是有效地生成测试用例。文中提出基于接口自动机(Interface Automata)技术自动生成列车通信设备的测试用例。现有的接口自动机技术只支持同步消息的发送与接收,而列车通信设备还需要处理异步消息的发送与接收。为此,文中对接口自动机理论进行了扩展,以便支持列车通信设备异步消息通信的需求。依据列车通信网络的规范,文中使用扩展了的接口自动机对列车通信模块及通信模块的组合作了形式化建模,依据测试覆盖准则,自动生成测试序列,并利用模型驱动技术自动生成测试脚本。文中比较了手工生成测试用例方法及所提出的自动生成测试用例技术,展示后者从测试覆盖、时间节约方面的有效性。

**关键词:**嵌入式系统;列车通信网络;接口自动机

中图分类号:TP18

文献标识码:A

文章编号:1673-629X(2013)05-0001-05

doi:10.3969/j.issn.1673-629X.2013.05.001

## Test Case Generation for Train Communication Devices Based on Interface Automata

YU Lian<sup>1</sup>,LI Jun-long<sup>1</sup>,BAI Xiao-ying<sup>2</sup>,LIU Xin<sup>1</sup>,WU Zhong-hai<sup>1</sup>

(1. College of Software and Microelectronics, Peking Univ., Beijing 102600, China;

2. Computer College, Tsinghua Univ., Beijing 100084, China)

**Abstract:** Train communication network device is a kind of embedded systems. Software testing has been still an effective way to assure the quality of train communication network device. One of the most critical tasks for embedded system testing is how to generate test cases effectively. Present a test-case generation approach that is based on interface automata. Currently, interface automata simply supports sending and receiving of messages synchronously, which is incompatible with the mode of message handling of train communication network. Therefore, propose a way to extend interface automata to support the demand of asynchronous message communication for train communication device; model the communication components of the train communication network and their interactions; with this model, generate test sequences according to test coverage criteria, and produce test scripts automatically. Finally, compare the approach proposed with manual test-case generation, and show the efficiency and the effectiveness in test coverage.

**Key words:** embedded systems; train communication network; interface automata

## 0 引言

列车通信网络(Train Communication Network,简称TCN),是一种以计算机网络为核心的分布式网络控制系统,作为铁路机车车辆的控制、检测和诊断系统。该国际标准由国际电工委员会(IEC)和国际铁路联盟(UIC)联合制定,即IEC61375<sup>[1]</sup>,同时电气电子工程师学会(IEEE)也引用该项标准作为列车通信网

络标准,即IEEE1473-T<sup>[2]</sup>。列车通信网络属于安全关键(safety-critical)系统,又称作生命关键系统,是指一旦系统发生错误或者故障,会导致人员的伤亡或者严重的设备损害或者环境伤害的系统<sup>[3]</sup>。因此,如何保证这类嵌入式系统的质量是目前软件测试领域急需解决的问题。

软件测试是质量保证的主要方法之一,被广泛地应用于嵌入式系统的测试中<sup>[3,4]</sup>。基于TCN标准规约的列车通信设备测试需要测试基于车辆总线及列车总线的通信。前期基于手工方式所进行的列车通信设备测试存在诸多明显问题。例如,缺乏规范的测试技术,测试用例生成效率低,测试覆盖率低。

收稿日期:2012-11-08;修回日期:2013-03-03

基金项目:国家自然科学基金资助项目(60973001);2012年“AutoTest软件系统测试”项目

作者简介:郁 莲(1963-),女,博士,从事软件测试理论及方法研究。

文中提出基于接口自动机技术 (Interface Automata)<sup>[4]</sup>, 依据列车通信的接口规约 (包括输入、输出等描述), 自动生成列车通信设备测试用例, 并对列车通信设备进行测试。接口自动机技术能够规范化地表达列车通信的接口规约, 并且可以验证通信模块之间的组合的正确性<sup>[5,6]</sup>。但是, 现有的接口自动机理论的假设前提是模块之间的通信是同步的, 不支持异步通信系统的建模。然而在列车通信设备中, 有异步通信需求。所以文中对接口自动机理论进行了扩展, 支持列车通信设备的异步消息通信建模需求。依据列车通信网络的规范, 文中使用扩展了的接口自动机技术对列车通信模块及通信模块的组合作了形式化建模, 依据测试覆盖准则, 自动生成测试序列, 并利用模型驱动技术自动生成测试脚本。

## 1 测试流程及测试实施架构

### 1.1 测试流程

测试流程包括列车通信设备需求分析, 列车通信模块规约的接口自动机建模, 测试驱动 (Test Driver) 模块接口自动机建模, 各模块接口自动机组合, 测试路径生成, 测试用例生成, 测试用例执行。

需求分析和被测系统接口自动机建模, 测试驱动接口自动机建模, 是人工完成的。对接口自动机组合, 测试路径自动生成和测试用例生成都实现自动化。对于接口自动机的建模和建模后代码生成的自动化部分, 采用模型驱动软件开发技术来实现。

### 1.2 测试实施架构

测试目标有两个模块, 一个是 MDFULL 端, 一个是 Server 端。测试是通过在 MDFULL 端调用应用层接口, 编写发送端 (接收端) 应用程序, 来与 Server 端的接收方 (发送方) 应用程序进行通信。从而测试应用层接口的正确性。具体的测试实施架构如图 1 所示。

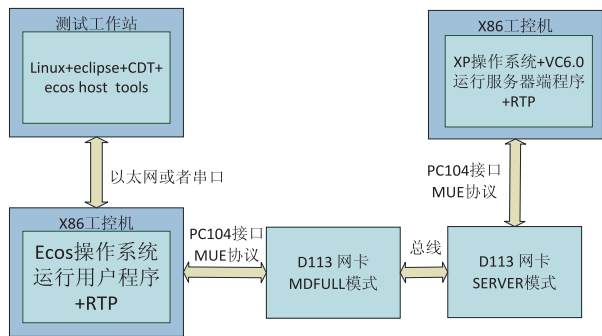


图 1 测试实施架构图

## 2 基于接口自动机测试用例生成

### 2.1 扩展的接口自动机的基本概念

接口自动机的形式化定义如下<sup>[7]</sup>:

定义 1 接口自动机: 将接口自动机  $P$  形式化表示为一个六元组,  $P = \langle V_p, V_p^{\text{init}}, A_p^I, A_p^O, A_p^H, T_p \rangle$ , 其中元素说明如下<sup>[3]</sup>:  $V_p$  表示状态集合, 单个状态用  $v_i$  表示,  $v_i \in V_p$ ;  $V_p^{\text{init}}$  为初始状态集合, 每个接口自动机至少包含一个初始状态, 如果  $V_p^{\text{init}} = \emptyset$  则  $P = \emptyset$ , 即该接口自动机为空。  $A_p^I, A_p^O, A_p^H$  分别表示输入, 输出和内部动作集, 动作集两两不相交,  $A_p = A_p^I, A_p^O, A_p^H$ 。单个输出动作作用“!”来标识, 输入动作作用“?”标识, 内部动作使用“;”来标识。

动作同步/异步: 对于可组合的两个接口自动机  $P$  和  $Q$ , 如果组合的时候, 存在共同动作  $a$ , 能够在某个状态下,  $P$  的输出动作“ $a!$ ”与  $Q$  的输入动作“ $a?$ ”能够转换为  $a$ ; 则称  $a$  在状态下动作同步; 否则称动作异步。动作同步或者异步只有在接口组合后才能表现出来。

可异步动作: 如果状态  $v$  可以在  $a$  同步的情况和  $a$  异步的情况下转换到  $v'$ , 则称  $a$  为可异步动作, 相反的, 如果状态  $v$  只能在  $a$  同步的情况下转换到  $v'$ , 则称  $a$  为不可异步动作。

带条件的动作的标识: 假设  $a', a \in A_p$ , 其中  $a$  是可异步动作,  $v, u, w \in V_p$ , 存在子序列  $(v, a, u, a', w)$ 。如果  $a$  动作同步时, 动作  $a'$  发生, 则动作  $a'$  标识为  $\{a; a'\}$ ; 如果  $a$  动作异步时, 动作  $a'$  发生, 则动作  $a'$  标识为  $\{\neg a; a'\}$ 。其中  $a'$  称为带条件的动作。

$T_p \subseteq V_p \times A_p \times V_p$  表示状态的转换集合,  $(v, a, v') \in T_p$ , 记为  $v \xrightarrow{a} v'$ 。

组合接口自动机的形式化定义如下:

定义 2 接口自动机组合条件: 如果接口自动机满足条件:  $A_A^I \cap A_B^I = A_A^O \cap A_B^O = \emptyset, A_A^H \cap A_B^H = A_B^H \cap A_A^H = \emptyset$ , 则称  $A, B$  两个接口自动机可组合。另外记  $\text{shared}(A, B) = A_A \cap A_B$ 。如果两个接口自动机可以组合, 则  $\text{shared}(A, B) = (A_A^I \cap A_B^O) \cup (A_A^O \cap A_B^I)$ 。

定义 3 接口自动机乘积: 如果  $P$  和  $Q$  能够组合, 它们的组合接口自动机概念有如下定义:

$$V_{P \times Q} = V_P \times V_Q, V_{P \times Q}^{\text{init}} = V_P^{\text{init}} \times V_Q^{\text{init}},$$

$$A_{P \times Q}^I = A_P^I \cup A_Q^I \setminus \text{shared}(P, Q),$$

$$A_{P \times Q}^O = A_P^O \cup A_Q^O \setminus \text{shared}(P, Q),$$

$$A_{P \times Q}^H = A_P^H \cup A_Q^H \cup \text{shared}(P, Q)$$

对于  $a \in \text{shared}(P, Q)$ , 采用  $\neg a$  表示内部异步发生的内部消息,  $a$  表示同步发生的内部消息。

$$((v_A, v_B), a, (v'_A, v'_B)) \in T_{A \times B}, \text{ 如果}$$

$$a \notin \text{shared}(A, B) \wedge (v_A, a, v'_A) \in T_A \wedge v_B = v'_B;$$

$$a \notin \text{shared}(A, B) \wedge (v_B, a, v'_B) \in T_B \wedge v_A = v'_A;$$

$$a \in \text{shared}(A, B) \wedge (v_A, a, v'_A) \in T_A \wedge (v_B, a, v'_B) \in T_B;$$

$((v_A, v_B), \neg a, (v'_A, v'_B)) \in T_{A \times B}$ , 如果  
 $a \in \text{可异步动作} \wedge a \in \text{shared}(A, B) \wedge (v_A, a, v'_A) \in T_A \wedge v_B = v'_B$ ;  
 $a \in \text{可异步动作} \wedge a \in \text{shared}(A, B) \wedge (v_B, a, v'_B) \in T_B \wedge v_A = v'_A$ ;

## 2.2 被测模块的接口自动机建模

### 2.2.1 通信系统单个接口描述

发送端接收一个 call\_request 输入, 然后发出一个 send\_msg 消息。消息成功发送后, 在超时事件发生前, 如果接到接收端发送的回复消息的握手消息(ack), 则对这个握手消息进行回复(reack), 并且等待接收端的回复消息(reply\_msg), 当收到接收端回复的消息后, 系统调用 call\_confirm 回调函数作为系统的输出, 反馈给系统接口的调用者。如果在 send\_msg 消息已被接收端接收后, 并且消息发送端未接收到接收端回复的 ack 消息前, 超时事件发生, 系统内部发出一个超时信号(time\_out), 根据定义, 将该状态转移边标记为 {send\_msg; time\_out}, 并且调用 call\_confirm 回调函数作为系统的输出, 反馈给系统接口的调用者。如果 send\_msg 消息未被接收端成功接收, 发送端超时, 系统内部发出一个超时信号(time\_out), 根据定义, 将该状态转移边标记为 {¬ send\_msg; time\_out}。消息发送端的消息发送接口的接口自动机描述如图2所示。

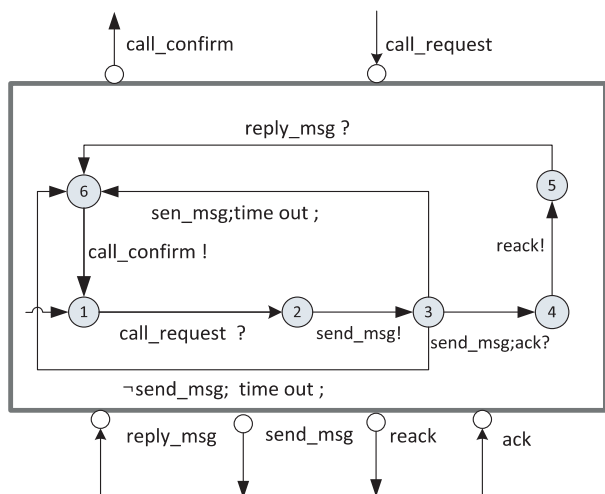


图2 消息发送接口的接口自动机模型

被测系统中接收端又分为两个接口, 消息接收接口和消息回复接口。消息接收接口接收一个 receive\_request 输入, 如果输出一个 receive\_ok, 则继续等待发送端发出的 send\_msg 消息, 当接收到 send\_msg 消息后, 系统调用 receive\_confirm 回调函数作为系统输出。当接收 receive\_request 输入后, 如果输出 receive\_ok 消息, 则系统重新回到等待 receive\_request 系统输入的状态。消息接收模块的接口自动机模型如图3所示。

消息回复接口接收一个 reply\_request 输入, 然后向消息发送端发送一个握手消息(ack), 如果 ack 消息

成功发送, 发送端的握手反馈消息(reack), 状态转移边标记为 {ack; reack?}, 则输出一个 reply\_ok 消息, 并且发送一个 reply\_msg 给消息发送端, 然后系统将输出 reply\_confirm 回调函数作为系统的输出。如果握手消息 ack 未成功发出, 则系统输出一个 reply\_fail 消息, 状态转移边标记为 {¬ ack; reply\_fail}, 系统回到等待接收 reply\_request 输入的初始状态。接口自动机如图4所示。

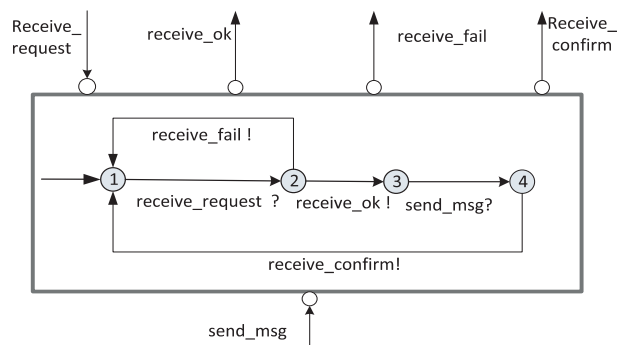


图3 消息接收接口的接口自动机模型

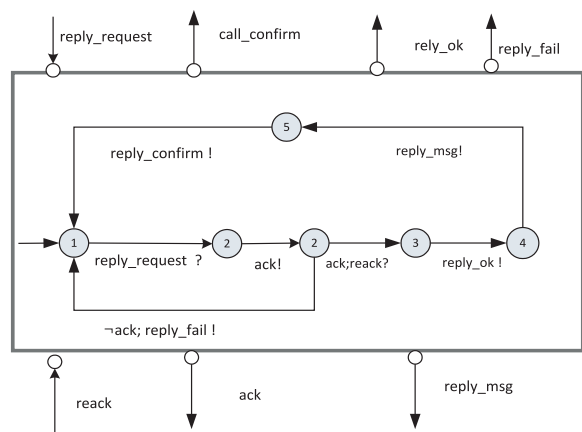


图4 消息回复接口的接口自动机模型

发送端测试驱动: 发送端的测试驱动对发送端进行初始化, 然后通过调用消息发送接口来向特定的接收端发送消息, 并且通过消息发送接口输出的回调函数来判断消息是否成功发送。

接收端测试驱动: 接收端的测试驱动对接收端进行初始化, 然后通过调用消息接收接口来接收发送端发送的消息, 并通过接收消息接收接口的回调函数判断消息是否成功接收。在消息成功接收的情况下, 调用消息回复接口, 向消息发送端回复消息, 并通过回复消息接口的回调函数来判断消息是否成功回复。

### 2.2.2 通信系统组合接口描述

本例中接口自动机组合通过三步进行。

(1) 发送端的测试驱动和消息发送接口进行接口组合, 得到消息发送端的接口自动机。

(2) 接收端的测试驱动和消息接收接口以及消息回复接口进行组合, 得到消息接收端的接口自动机。





代码生成的时候,超时动作对应生成为线程的休眠函数。

### 3 讨 论

对比手工生成的测试用例而言,通过接口自动机建模、组合,最后生成测试用例的方法能够提高列车通信设备测试的覆盖率,同时提高测试用例生成的效率。

对于文中示例采用人工测试的方法,只是覆盖了1,10两条路径。这两条路径都不能对发送端发生超时事件时,接收端回复函数被调用的返回值是否正确情况进行测试。在组合模型生成的10条路径中,路径2,3,4,5都对应发送端发送消息后,在接收端成功接收了消息,还未调用回复函数时,发送端发生了超时事件。测试需要验证发送端发生超时事件后,接收端回复函数被调用时的返回值是否正确。所以相对于手工测试,采用了接口自动机模型来生成测试路径提高了测试的覆盖率<sup>[10]</sup>。

手工测试方法中,对于路径1设计和编码测试了100个左右的测试用例。即使采用覆盖率最低要求的状态覆盖,需要10条测试路径,将要生成不少于1000个测试用例。要将整个列车通信网络协议的接口进行测试,则至少需要几万个测试用例。如果文中采用覆盖率更高的转移覆盖策略或者状态转移组合覆盖策略,则会产生更多的测试路径和测试用例。

根据本项目实验数据的估算,手工生成测试用例需要对领域知识的理解然后直接进行手工代码编写,生成测试用例平均用时大约为10~30分钟。总的生成测试用例时间与测试用例成正比。对于自动生成测试用例而言,首先需要开发测试工具,工作量约为2~3个月。利用工具对被测的接口进行规范化的建模,工作量约为20分钟到1个小时,然后是测试用例代码自动生成。所以当需要生成测试用例个数较少(几~几十个)时,平均每个测试用例的生成时间可能达到2到3个月。但是对于需要生成大量(成千上万)测试用例时,平均每个测试用例的生成时间随着用例个数的增加而急剧缩小,最后可以稳定到毫秒级的水平。而当系统需求规约发生变化时,自动化测试技术只需修改规约模型,便可瞬时生成测试用例,从而大大节约测试人力资源和时间<sup>[11]</sup>。

### 4 结束语

文中从TCN协议的测试需求入手,采用模型驱动测试技术对列车通信设备进行自动化测试。提出了基于接口自动机技术对被测系统进行规范建模,并对接口自动机作了扩展以支持异步消息的建模。为了实现自动化的测试,使用模型驱动技术,采用Eclipse的模型驱动开发插件JET和Acceleo对整个测试流程自动化<sup>[12]</sup>。文中较为详细地讨论了对于TCN被测系统模块建模,模块组合建模,以及根据不同的覆盖策略生成测试路径,自动化生成测试路径的输入参数及可执行脚本。与前期手工测试方法比较,文中所提出的方法明显地提高了TCN列车通信设备测试覆盖率及测试效率。在以后的工作中,将把文中提出的方法及技术应用到更多TCN模块测试,进一步验证基于接口自动机测试用例自动生成方法及技术的有效性。

#### 参考文献:

- [1] Kirmann H, Zuber P A. The IEC/IEEE Train Communication Network[J]. IEEE Micro, 2001, 21(2): 81-92.
- [2] 王晓燕, 刘淑芬, 于海. 基于接口自动机的服务组合方法[J]. 吉林大学学报, 2009, 39(3): 743-748.
- [3] 王丽, 赵卢霞, 李卫红, 等. 嵌入式软件测试工程化研究与实践[J]. 计算机应用, 2009, 29(S2): 192-194.
- [4] 牛为华, 孟建良, 张素文. 成对测试中的测试生成策略研究[J]. 计算机仿真, 2003, 20(7): 111-113.
- [5] 简林. 基于接口自动机的服务组合验证研究[D]. 长沙: 中南大学, 2011.
- [6] 刘利枚, 汪文勇, 唐科. 嵌入式软件测试方法与技术[J]. 计算机与现代化, 2005(4): 123-126.
- [7] de Alfaro L, Henzinger T A. Interface Automata[J]. ACM SIGSOFT Software Engineering, 2001, 26(5): 4-10.
- [8] Kirmann H. Train Communication Network IEC 61375 - 4 Wire Train Bus[D]. Ecole: Ecole Polytechnique Fédérale de Lausanne (EPFL), 1999.
- [9] Law A, Keton W D. Simulation Modeling and Analysis[M]. New York: McGraw Hill Book Company, 1982.
- [10] 胡军, 于笑丰, 张岩. 基于场景规约的构件式系统设计与验证[J]. 计算机学报, 2006, 29(4): 500-510.
- [11] 殷永锋, 王轶辰, 余正伟. 嵌入式软件测试技术[M]. 北京: 北京航空航天大学出版社, 2005: 20-31.
- [12] 熊光泽. 嵌入式系统软硬件协同设计技术综述[J]. 计算机应用, 2006, 26(4): 757-760.

基于接口自动机的列车通信设备测试用例生成

作者：郁莲，李军龙，白晓颖，刘鑫，吴中海

作者单位：郁莲, 李军龙, 刘鑫, 吴中海 (北京大学 软件与微电子学院, 北京 102600)，白晓颖 (清华大学 计算机学院, 北京 100084)

刊名：计算机技术与发展

英文刊名：Computer Technology and Development

年，卷(期)：2013 (5)

本文链接：[http://d.g.wanfangdata.com.cn/Periodical\\_wjtz201305003.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjtz201305003.aspx)