

AUTOSAR 标准一致性测试研究

谭文志, 邝继顺, 王德志, 李仁发

(湖南大学信息科学与工程学院, 长沙 410082)

摘要: 分析 AUTOSAR 一致性测试过程, 采用 eclipse、TTthree、 μ TTman 等工具, 基于 TTCN-3 测试系统架构, 实现 AUTOSAR 一致性测试。针对简单信号灯系统, 研究测试系统中测试套、编解码器和适配器的功能及相关实现方法, 并验证采用 TTCN-3 进行 AUTOSAR 一致性测试的可行性。

关键词: AUTOSAR 标准; TTCN-3 标准; 一致性测试; 测试套; 适配器

Research on AUTOSAR Conformance Test

TAN Wen-zhi, KUANG Ji-shun, WANG De-zhi, LI Ren-fa

(School of Information Science and Engineering, Hunan University, Changsha 410082, China)

【Abstract】 It analyzes the AUTOSAR conformance test process. The AUTOSAR conformance test based on TTCN-3 test system architecture has been achieved using the tools such as eclipse, TTthree and μ TTman. The functions of test suite, codec and adapter of testing system and their implementation have been discussed for simple blinker system. The article verifies the feasibility of AUTOSAR conformance test using TTCN-3.

【Key words】 AUTOSAR; TTCN-3; conformance test; test suite; adapter

DOI: 10.3969/j.issn.1000-3428.2012.04.014

1 概述

AUTOSAR 是汽车开放式系统架构(Automotive Open System Architecture)的简称, 定义了一套支持分布式、功能驱动的汽车电子软件开发方法和电子控制单元(Electronic Control Unit, ECU)上的软件架构标准化方案, 以便应用于不同的汽车和平台, 提高软件复用, 降低开发成本^[1]。AUTOSAR 标准提出采用一致性测试来降低电子控制单元间协作风险, 一致性测试是汽车安全的一个重要保障。TTCN-3(Testing and Test Control Notation)是一种类程序设计语言, 本文采用 TTCN-3 编写测试套, 完成对 AUTOSAR 标准的一致性测试。

2 AUTOSAR 介绍

AUTOSAR 的目标是实现汽车电子基本系统功能以及功能接口的标准化, 使得功能易于修改和集成, 提高软件的升级能力, 实现应用软件组件与基础软件的分离, 从而能够独立于底层的硬件。

AUTOSAR 架构可以分为 3 个层次: 顶层是应用层(application layer), 是应用软件组件和执行器软件组件的集合, 在应用层, 软件的结构类型是组件式的; 底层是微控制器硬件层, 与实际微控制器连接, 映射微控制器的功能和外围接口, 定义了内存接口、通信接口以及 I/O 驱动接口; 中间层包括 AUTOSAR 运行时环境(Run Time Environment, RTE)和 AUTOSAR ECU 基础软件, 是独立于硬件的应用软件集成平台。RTE 为应用层软件组件提供通信服务, 是 ECU 内部或者与外部 ECU 之间进行信息交换的通信中心, 是提供软件复用的基础。基础软件是标准化的软件层, 包含了标准化和 ECU 特定的组件。

基础软件本身也可以分为多个层次, 自底向上可以分为微控制器抽象层、ECU 抽象层和服务层, 还有跨越 3 个层次的复杂驱动层^[2-3]。整个 AUTOSAR 体系架构是层次化、模块化的结构。AUTOSAR 体系结构如图 1 所示。

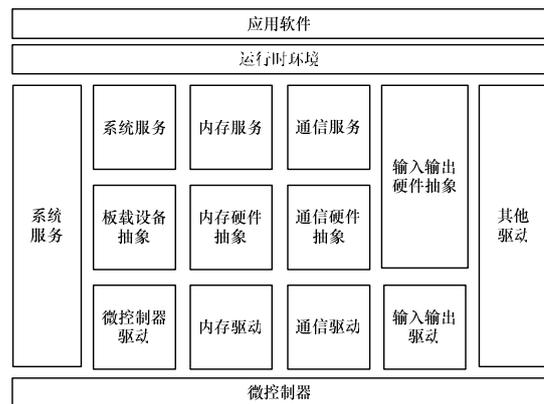


图 1 AUTOSAR 体系结构

软件组件(Software Component, SWC)是 AUTOSAR 中一个重要概念。它封装了全部或者部分汽车电子功能模块, 包括其具体的功能实现和对应的描述。软件组件的实现独立于其所在的微控制器和 ECU 类型, 它的独立性是通过虚拟功能总线(Virtual Functional Bus, VFB)来实现的。通过虚拟功能总线, 软件组件之间的通信细节被抽象出来。软件组件通过 AUTOSAR 定义的接口对通信进行描述, 实现与其他软件组件和硬件的交互。软件组件都需要定义端口, 端口代表了软件组件间通信内容和方向。

每个软件组件都提供了其代码的具体实现以及描述文件。代码实现即运行实体的实现, 采用 C 源文件方式提供。描述文件用来描述组件的属性, 包括所使用的端口、端口接

基金项目:“核高基”重大专项“实时嵌入式操作系统及开发环境”(2009ZX01038-001-008)

作者简介: 谭文志(1984—), 男, 硕士研究生, 主研方向: 软件测试; 邝继顺, 教授; 王德志, 硕士研究生; 李仁发, 教授

收稿日期: 2011-08-12 **E-mail:** tanwenzhip@yaho.com.cn

口、运行实体、以及运行实体等，以 XML 文件形式提供。

3 TTCN-3 介绍

TTCN-3 是欧洲电信联盟发布的一种抽象测试描述语言，它主要用于描述在多种通信端口上的各种响应系统测试，应用的典型领域是协议测试(包括移动和互联网协议)、服务测试、基于平台的 CORBA 测试、API 测试等^[4]。

TTCN-3 支持多种表示格式，分别是核心语言格式、表格表示格式、图形表示格式和用户自定义格式，如图 2 所示。针对不同的应用，用户可以选择不同的表现格式来描述测试套。TTCN-3 核心语言是 TTCN-3 语言定义的一种文本表现格式，采用模块化的结构，具有良好的可读性。核心语言可以作为基于 TTCN-3 的测试执行工具和测试开发工具间的一种标准交换格式或是作为其他表现格式的语义基础，AUTOSAR 类型可以很好的转换到 TTCN3 类型。本文设计的测试套采用的是核心语言格式。TTCN-3 规范提出了 TTCN-3 测试系统结构，如图 3 所示。

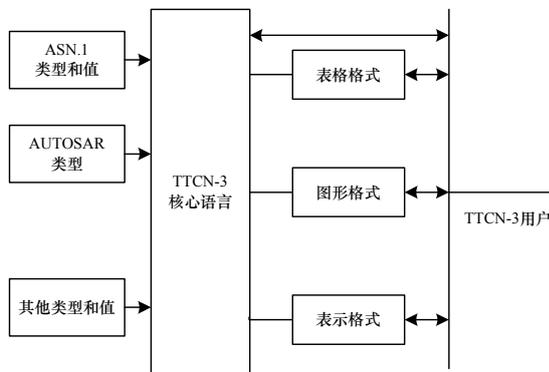


图 2 核心语言和各种表示格式

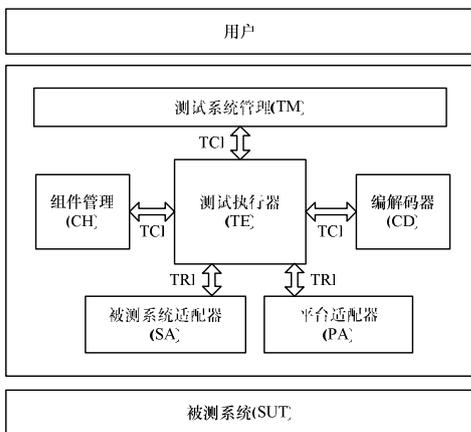


图 3 TTCN-3 测试系统结构

TTCN-3 测试系统结构主要分为 TM(测试系统管理)、CD(编解码)、CH(组件管理)、TE(测试执行器)、SA(被测系统适配器)、PA(平台适配器)、SUT(被测系统)。

(1)TM: TM 实体负责测试系统的全面管理。

(2)CD: CD 实体负责外部编解码工作。编解码数据与 TE 内部基于消息或过程的通信有关。

(3)CH: 一个 TTCN3 测试用例包含的测试组件通常有多个，CH 实体负责管理多个测试组件。CH 实体允许 TM 创建和控制分布测试系统，这种方式独立于 TE。

(4)TE: TE 实体负责解释和执行 TTCN-3 测试用例。从概念上来说，TE 可以分为：ETS(可执行测试套)，T3RTS (TTCN-3 运行时系统)，EDS(内部编解码系统)。这些实体通

过 TRI 接口与外部交互。

(5)SA: SA 用于处理 TE 与 SUT 之间的消息通信和过程调用。

(6)PA: PA 是与操作系统密切相关的功能模块，在测试系统中，PA 用于协助测试执行器实现外部函数和定时器功能。

另外，TCI(TTCN-3 Control Interface)是系统的控制接口，TRI(TTCN-3 Runtime Interface)是系统的运行接口，通过这 2 个接口可以实现测试套与被测系统进行通信，从而实现测试目的^[5]。

4 AUTOSAR 一致性测试系统设计与实现

AUTOSAR 一致性测试实现主要分为 3 个步骤：第 1 步是测试生成，即分析有关的软件说明(Software Specification, SWS)，提取与一致性测试相关条目，选择测试方案。第 2 步是测试实现，根据测试对象设计 TTCN3 测试用例，完成抽象测试套。第 3 步是测试执行，将测试套运行在仿真的 BSW 模块上，验证软件组件是否符合 AUTOSAR 标准，并进行判定^[6-7]。

下面主要介绍 AUTOSAR 一致性测试过程中与 TTCN-3 相关部分，包括 AUTOSAR 数据类型与 TTCN-3 数据类型的转换，TTCN-3 测试套实现以及编解码和适配器实现。

4.1 AUTOSAR 数据与 TTCN-3 数据的转换

AUTOSAR 标准定义了一些数据类型用来描述 BSW 模块源代码，为了描述特定的目标系统属性，可以进一步转换为 C 语言数据类型。测试系统和平台适配器相关 AUTOSAR 数据类型必须转换为 TTCN3 数据类型。AUTOSAR 的数据类型可以分为简单数据类型和结构数据类型。表 1 给出了 AUTOSAR 基本数据类型转换到 TTCN-3 数据的对应关系。

表 1 基本数据类型转换

AUTOSAR 基本类型	TTCN-3 定义
UInt8	Type integer uint8 (0...255)
UInt16	Type integer uint16 (0...65535)
UInt32	Type integer uint32 (0...4294967295)
Sint8	Type integer sint8 (-128...127)
Sint16	Type integer sint16 (-32768...32767)
Sint32	Type integer sint32 (-2147483648...2147483647)
Float32	Type integer float32 (-3.4E38...3.4E38)
Float64	Type integer float64 (-1.79769E308...1.79769E308)

在 AUTOSAR 标准中，结构数据类型有 Record 类型和数组数据类型，采用 TTCN-3 中 record 和 record of 进行转换。举例如下：

AUTOSAR 中 record 类型为：

```

ApplicationError:
Type: record
Element: CompID (INTEGER)
          ErrorID (INTEGER)

```

转换为等价的 TTCN-3 类型为：

```

Type integer CompID(0..31)
Type integer ErrorID(0..255)
Type record ApplicationError(comp compID,_error ErrorID)

```

4.2 测试套设计

下面以一个基于 AUTOSAR 标准的简单信号灯系统为

例, 介绍测试套的实现。信号灯系统模型如图4所示。

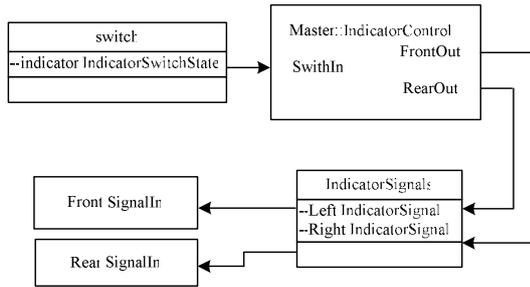


图4 简单信号灯系统模型

简单信号灯系统包含一个控制器(Master)和多个指示器组件。控制器相当于一个简单的ECU, 其功能是接收来自switch的输入信号, 生成输出信号给汽车前后灯指示器。控制器有3个端口, 一个输入端口SwichIn, 用于接收开关(switch)的信号, 开关信号分为left, right和off3种类型。另外有2个输出端口FrontOut和RearOut, 控制器通过这2个端口发送信号到信号组件指示器(IndicatorSignals), 指示信号分为on和off2种类型, 最后指示器将信号传送给对应的执行器(Front和Rear)。将对这个模型中指示灯指示器能否准确接收到控制器发送的信号这个过程进行测试。可以将这个模型端口和信号用TTCN-3语言描述为:

```
module SimpleModule{
    type enumerated Signal {on,off};
    type enumerated Switchstate {left,right,off};
    type port Signals message {in Signal};
    type port Sswitch message {out Switchstate};
    type component TestComponent{
        port Signals FrontOut, RearOut;
        port Sswitch SwichIn;}
}
```

在设计的测试用例中, 假设SwichIn端口发送信号left, 如果FrontOut端口和RearOut端口在30ms内收到信号on, 则测试验证结果为pass, 而FrontOut端口和RearOut端口收到其他信号或者超时, 验证结果为fail。测试用例如下:

```
testcase tc() runs on TestComponent{
    ...
    var datetime t_time;
    const timespan t_max =30*millisec;
    SwichIn.send(left)->timestamp t_time;
    alt{
[]FrontOut.receive(on) within (t_time .. t_time+t_max){
    RearOut.receive(on) within ((t_time .. t_time)+t_max);
    Setverdict(pass);}
[]RearOut.receive(on) within ((t_time .. t_time)+t_max){
    FrontOut.receive(on) within ((t_time .. t_time)+t_max);
    Setverdict(pass);}
[]RearOut.receive(){ Setverdict(fail); }
[]FrontOut.receive() {Setverdict(fail); }
[] RearOut.catch(timeout) {setverdict(fail); }
[] FrontOut.catch(timeout) {setverdict(fail); }
}
```

测试用例完成后, 增加测试控制部分。测试控制部分负责控制测试用例的执行顺序, 这里只设计了一个测试用例, 所以只需要往测试控制部分加入execute(tc())语句, 执行这个测试用例, 这样就完成了抽象测试套的设计。TTCN-3抽象

测试套完成后, 还需要将它通过TTCN-3编译器生成可执行测试套。本文选择采用TTthree作为TTCN-3编译器, TTthree是德国Testing tech公司开发的开源编译器, 将TTCN-3抽象测试套编译并且打包为一个jar文件的可执行测试套, 同时生成了一个模块加载文件。该模块加载文件是一个符合XML语法规则的配置文件, 文件中包含被测系统适配器和测试用例相关信息。

4.3 编解码器与被测系统适配器

编解码器负责对TTCN-3测试数据进行编码, 同时还对接收数据进行解码。它需要实现TTCN-3标准中定义的TciCDPProvided接口, 在本文的例子中, 编解码函数继承encode()和decode()2个方法进行实现。

被测系统适配器负责测试系统与测试系统进行消息通信, 需要实现TTCN-3运行时接口(TTCN-3 Runtime Interface, TRI)操作中相应的方法。其中, triExecuteTestCase()初始化测试系统接口端口, 建立与被测系统的连接。TriMap()和TriUnmap()负责通信端口的映射和关闭。TriSend()发送测试数据到被测系统。triEnqueueMsg()从被测系统接收数据, 通过相应端口发送给测试套。上述编解码和适配器均在Eclipse平台下用Java语言实现, 然后将编解码器与被测系统适配器打包为jar文件。

4.4 测试运行

把可执行测试套、编解码器和被测系统适配器都放到μTTman平台上运行, μTTman是一个测试管理工具, 可以支持TTCN-3的灵活执行。μTTman平台根据用户提供的被测系统适配器, 建立起被测系统与可执行测试套之间的通信。在试验中, 被测系统需要仿真实现, 在这里不对被测系统进行介绍。最终μTTman平台给出测试结果pass或者fail。

5 结束语

AUTOSAR标准在全球范围已经得到越来越广泛的应用, 而一致性测试是该标准不可缺少的部分。本文在对AUTOSAR一致性测试研究中, 完成AUTOSAR数据类型到TTCN3数据类型的转换, 采用TTCN3来编写测试套, 对被测系统进行测试。下一步工作是集成一个进行AUTOSAR一致性测试的运行环境, 从而提高测试效率。

参考文献

- [1] Welcome to the AUTOSAR Development Partnership[EB/OL]. [2011-08-01]. <http://www.autosar.org>.
- [2] AUTOSAR BSW & RTE Conformance Test Specification Part 1: Background R4.0[Z]. 2010.
- [3] AUTOSAR BSW & RTE Conformance Test Specification Part 2: Process Overview R4.0[Z]. 2010.
- [4] European Telecommunications Standards Institute. Methods for Testing and Specification(MTS): The Testing and Test Control Notation[EB/OL]. (2009-06-21). http://t-ort.etsi.org/file_download.php?file_id=2173&type=bug.
- [5] 李华, 张巨萍. TTCN-3测试系统协议相关部分的分析与实现[J]. 计算机工程, 2009, 35(21): 253-255.
- [6] AUTOSAR BSW & RTE Conformance Test Specification Part 3: Creation & Validation R4.0[Z]. 2010.
- [7] AUTOSAR BSW & RTE Conformance Test Specification Part 4: Execution Constraints R4.0[Z]. 2010.