

一种 TTCN 测试套自动生成方法研究

陈 萍 赵会群 尚思超

(北方工业大学信息工程学院, 北京 100041)

摘 要: 简要介绍了 TTCN-3 标准的第三部分 GFT(Graphical presentation Format, 图形表示格式), 并且根据 GFT 与 TTCN-3 核心语言的内在关联以及标准中的语法规则设计开发了一套由图形表示格式自动生成用核心语言描述的测试套的转换工具, 从而可以减少测试相关人员在测试套开发过程中的工作量。

关键词: TTCN-3; GFT; 测试套; 测试用例

中图分类号: TP393

引 言

随着软件工程的发展, 软件测试技术越来越受到重视。在实现系统测试时, 一般由测试人员编写多个测试套(test suite)让系统运行, 然后得出系统能否准确工作的结论。测试套是测试用例(test case)和测试控制(test control)的统称, 一个测试套至少包括一个测试用例, 若有多个测试用例存在, 执行时就有一个顺序问题。用程序语言描述这些测试用例往往使测试人员较难直观得到测试用例执行顺序的消息, 而用图形格式描述执行过程就能使测试过程展现得更加直观, 从而使测试套的读写更直观易懂。因此本文中提出了一个从 GFT 到 TTCN-3 核心语言转换的工具 TTShow, 通过这个工具可以使测试套编写更为方便快捷, 也大大减少了测试人员编写测试用例的工作量。

此外, 目前许多开发工具都嵌入了 SDL 和 MSC 这样的形式化描述语言, 因此对形式化语言规范和 TTCN-3 测试套之间交互的研究就变得必要, 这同样使测试用例的自动生成机制变得可能^[1]。

1 TTCN-3 标准的图形表示格式 GFT

在一致性测试过程中, 测试实例的核心行为是测试系统与被测系统以及测试组件间的互动。GFT

可以清晰地表示组成部分和它们之间的互动消息, 从而支持测试组件间的图形差异与被测系统的通信连接及测试组件之间的通信。并且被测系统及测试系统内部的消息交换也可以更好地表现出来。

根据 ISO/IEC 9646-3 中关于一致性测试方法的定义, 测试是由测试目的决定的。为了更好地实现这些目的, 在抽象测试套中通常包含一个或更多的抽象测试用例, 它们定义了测试过程的必然行为。GFT 中引入形式语言 MSC (ITU-T Z. 120)^[2]的主要目的是用图形化语言描述测试目的和抽象测试用例。相应于这两种不同的描述对象, 对应两种不同的应用领域。

(1) 从对测试目的的图形化描述生成抽象测试用例。TTCN-3 核心语言或 GFT 都可以用于表示抽象测试用例。然而由测试目的生成测试用例是一个复杂的过程, 该过程同时也涉及到了 CATG (Computer Aided Test Generation) 技术的应用和发展。

(2) TTCN-3 图形表示格式 GFT 的发展以及 GFT 与 TTCN-3 之间映射关系的定义^[3]。

GFT 可借助图形表示抽象测试套的行为, 这部分包括模块控制部分、测试用例、函数和可选步, 但 GFT 不为相关的数据说明如类型声明、模版或 TTCN-3 模块结构提供图形化表示。具体来说, GFT 不提供图形化表示的部分包括模块参数定义、引用定义、类型定义、签名声明、模版声明、常量声明、外部常量声明和外部函数声明。虽然这部分定义和声明没有对应的 GFT 表示, 但可以用核心语言 (Core Language) 或者表格表示格式 TFT (Tabular presentation Formatt, ES 201 873-2) 进行表示^[3]。

收稿日期: 2007-04-27

基金项目: 北京市自然科学基金(4062012)

第一作者: 女, 1977 年生, 硕士生

E-mail: cityicefox@163.com

因为 GFT 是基于 MSC 描述语言的,从而 GFT 图形和 MSC 图形的对应部分可以相互对照,但为了更好的描述测试过程,在 GFT 中也采用了一些新的图形符号,主要有端口实例的表述、测试组件的创建、测试组件的开始、函数调用的返回、基于过程调用的时间管理、测试用例的执行、错误的激活和挂起、标签和跳转、调用中的定时器等。图 1 给出了 GFT 图形格式表示的测试用例 My TestCase 与使用核心语言直接进行描述之间的对照。

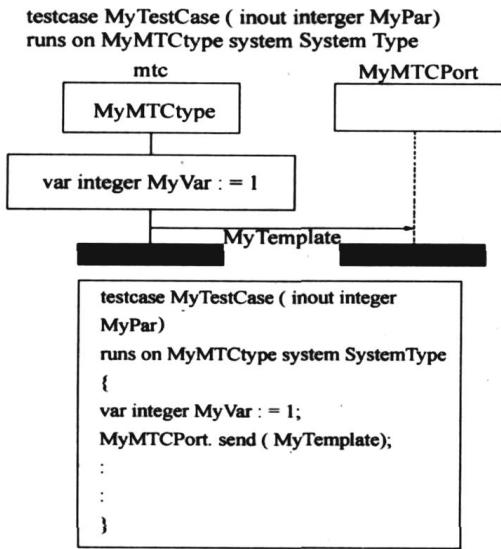


图 1 用 GFT 描述的测试用例图形和相应的核心语言比较

Fig. 1 Principle shape of a GFT test case diagram and corresponding core language

2 工具设计

TTCN-3 核心语言和一致性测试联系紧密,它除了作为独立的测试语言,还可作为测试套的标准化互换格式及各种表示格式的语义语法基础^[4]。

本文开发的这套工具 TTShow 从整体上讲是由 GFT 图形编辑模块和图形/代码转换模块两部分组成。其中 GFT 图形编辑模块负责为用户提供一个界面友好易操作的 GFT 图形编辑平台,在这个编辑平台上用户可根据自己的实际需求及相关标准自行绘制测试套图例,并且能实现用户对图形的自由缩放与拖拽功能、支持撤销与重做功能以及属性编辑器和大纲缩略视图等编辑管理功能。

在图形/代码转换模块中则由系统对用户绘制的 GFT 图进行分析,首先由图表类别约束机制判断该图所属的图表类别。根据 TTCN-3 标准第三部

分规定,可以使用 GFT 表示的图表具体可分为模块控制部分图表、测试用例图表、函数图表和可选步图表四种。然后由代码生成机制提取图中涉及到的所有 GFT 节点模型(如端口、动作、条件、内联表达式、返回、创建等),然后由不同节点间消息交互的顺序和具体条件,在获取 GFT 节点模型的同时获取相关消息和节点的所有参数、方法调用序列及其对应条件,将这些元素对照 GFT 标准提供的映射规则进一步划分为条件、循环、选择、函数调用、结果返回等实例中所涉及到的语言要素;最后结合核心语言规定的语法约束条件确定这些语言要素的相互关系,从而得到完整的测试用例。这样,通过测试套测试用例的图形化绘制生成,可以方便的在用形式化语言表示的图形和使用程序语言编写的代码间进行对比分析,从而确定测试用例在一次测试行为中的信息交互行为是否正确,使测试套的设计更加标准化、直观化和自动化。TTShow 工具的图形/代码转换模块部分的设计结构如图 2 所示。

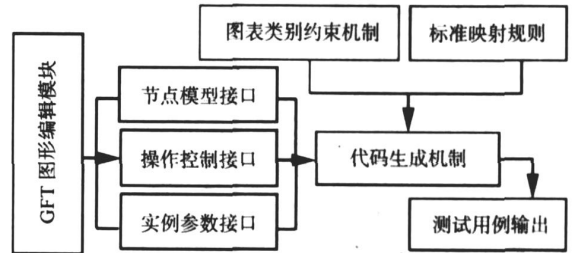


图 2 图形/代码转换模块设计结构

Fig. 2 The design structure of Diagram/Code conversion model

从整体上看,代码生成机制是图形/代码转换模块中的关键组成部分,最后由 TTShow 生成的测试用例都是通过代码生成机制完成从图形到代码转换的。当用户根据实际需求绘制完成相应的 GFT 图后,在工具中设计了一个用于实现图表类别约束机制的类 generateTTCN3Action,由该类对图表进行判断,作出属于模块控制、测试用例、函数或者可选步中的具体哪种类型的结论,从而确定对应代码的头部部分,默认情况下为测试用例。

在这里,生成代码的工作是由 TTCN3 Template 和 GenerateCode 两个类完成的。TTCN3 Template 类通过 generate() 方法实现了对实例代码主体部分(body)的架构,而 GenerateCode 类也是通过在 TTCN3 Template 类中对它的调用从而实现其代码转换功能的。在 GenerateCode 类中,本文定义了用

于生成 TTCN3 代码的 generateTTCN3()、生成内联表达式的 generateInlineExpression()、生成消息的 generateMessage() 和 generateOtherMessage() 方法, 以及用于调用在 GFT 图形编辑模块中定义的节点模型里面相对应的节点定义的方法 generateOther() 等。在这里, 因为每个图形节点在绘制的时候都有相应的节点模型(model)与之对应, 并且由节点模型和 TTCN-3 标准中第三部分规定的映射规则具体化(表 1 中给出了部分映射规则), 因此在生成代码时在 GenerateCode 类里可以根据不同模型对象类中定义的内容生成与其相匹配的代码。

表 1 GFT 格式的节点元素和核心语言之间的映射关系示例

Table 1 Mapping between GFT node symbols and corresponding TTCN-3 core language

节点模型	GFT 图形格式描述	核心语言描述
标签 label		label MyLabel;
跳转语句 goto		goto MyLabel;
选择语句 if-else		if(x>1){ MyPort.send(Template); }
while 循环语句		while(j<1){ MyPort.send(Template); }
函数顺序调用		MyFunction1(); MyFunction2();

比如对于 GFT 对象之间相互交换的消息实例 Message 来讲, 在生成代码时, 系统会调用 GenerateCode 类的 generateMessage() 方法, 再由 generateMessage() 根据模型中定义的规则来生成相应代码。这样最终的代码会由所有模型中的具体对象(如创建、连接、跳转、结束等等)来共同生成, 不同的对象对应不同的代码生成。

3 转换实例

以一个函数为例进行说明。在 TTCN-3 中函数用来表达和构造测试行为, 定义一个模块中的测试执行或组织计算, 函数可以返回一个值, 用关键字 return 来表示返回值^[2]。当 return 用在函数体中且返回类型定义在函数头部时, 其后要接一个常数、变量引用或一个表示返回值的表达式, 返回值的类型和返回类型应该是兼容的。函数体中的返回语句将会终止该函数的执行, 并且在函数调用的地方返回函数的返回值。

例子中的函数 MyFunction 借助 GFT 表示格式的描述见图 3。该函数头部定义为 function MyFunction (inout integer MyPar), 表明这是一个带有整形参数 MyPar 的函数, 其后又规定了函数返回一个整形值, 并使用端口 MyPTCtype 操作。在这个函数中, 我们为整形变量 MyVar 赋值, 且通过端口传送数据, 最后以函数形参 MyPar 和变量 MyVar 的和作为函数调用的返回值。

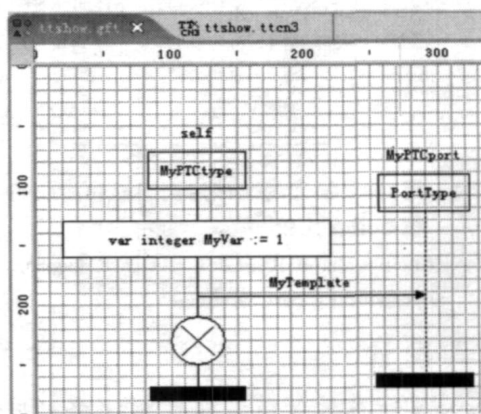


图 3 使用 GFT 格式表示的函数 MyFunction

Fig. 3 The GFT presentation of a TTCN-3 function MyFunction

在 TTShow 工具环境中, 对上述函数进行图形/代码转换之后所生成的 TTCN-3 模块代码如图 4 所示。当然在这里列举的仅仅是一个在结构上比较简单的函数, 但在 TTCN-3 标准定义下, 更为复杂的图形/代码的转换也是基于这个原理进行的。此外, 因为使用 GFT 图形表示格式不能从直观上对模块声明部分和定义部分加以表述, 如果要得到一个完整的测试套, 这部分代码依然是需要开发人员在转换工作完成后根据具体设计方案手工编写的。本文所设计的这个工具在测试过程中只是作为一个有

效的测试套辅助开发手段,以期可以减少相关人员在开发测试套过程中的工作量。

```

module ttshow :
function MyFunction (inout integer MyPar) return integer runs on MyPTctype (
var integer MyVar := 1;

MyPTCport.send(MyTemplate);
return MyVar+MyPar;
)

```

图 4 根据 GFT 图表生成的函数代码模块

Fig. 4 The code of Myfunction generated from GFT diagram

4 结束语

尽管 TTCN-3 在业界和研究领域已经获得了广泛应用,但关于 GFT 格式应用的研究相对较少。通常情况下都是将其作为一种测试辅助手段加以利用和描述,而将图形到测试套的自动生成应用在一致性测试过程中较少涉及。在本文提出的工具中,首先对用户根据实际情况绘制的 GFT 图形在图形标准规定的语法上加以分析,提取其中涉及的图形节点模型,然后根据这些模型间的消息交互关系及核心语言语法标准将节点进一步划分为方法、函数、输入、条件、输出等语言元素,最后由这些元素组合生成测试用例,从而加强了测试套的开发效率,在一定程度上实现了测试套开发的自动化。

不过因为基于图形生成测试套的方法还在不停的研究和发展中,而且通常将单个图形作为研究对象的情况居多,因此这套工具还存在一些需要进一步改进之处,比如对于 GFT 标准中规定的相同图形符号表示不同语义的情况下工具如何根据上下文自动识别,如何让这套工具在测试套开发过程与实际情况相结合以发挥更大的辅助作用,这些也正是下一步研究的主要方向。

参考文献:

- [1] GRABOWSKI J. TTCN-3-A new Test Specification Language for Black-Box Testing of Distributed Systems [C] Proceedings of the 17th International Conference and Exposition on Testing Computer Software (TCS '2000), Theme: Testing Technology vs. Testers' Requirements, Washington DC, 2000.
- [2] ITU-T Recommendation Z. 120. Message Sequence Chart (MSC) [M].
- [3] Methods for Testing and Specification (MTS) The Testing and Test Control Notation version 3 Part 3: TTCN-3 Graphical presentation Format(GFT) [S]. ETSI ES 201 873-3 V3. 1. 3, 2005.
- [4] Methods for Testing and Specification (MTS) The Testing and Test Control Notation version 3 Part 1: TTCN-3 Core Language[S]. ETSI ES 201 873-1 V3. 1. 1, 2005.

Study on automatic generation of TTCN test suite

CHEN Ping ZHAO HuiQun SHANG SiChao

(School of Information and Engineering, North China University of Technology, Beijing 100041, China)

Abstract: This paper introduces the third part of TTCN-3 standard- GFT (Graphical present Format). According to the relationship of GFT and TTCN-3 core language, and to the standard of syntax, we present the design and implementation of a Diagram/ Code automatic conversion tool called TTShow. With this tool, it can enhance work efficiency in the process of test suite design.

Key words: TTCN-3; GFT; test suite; test case