

U2TP 到 TTCN-3 自动转换的研究与实现

侯晓媛 王显荣 李 华 沈维维

(内蒙古大学计算机学院 呼和浩特 010020)

摘 要 在系统建模阶段使用 UML 2.0 Testing Profile (U2TP)集成测试信息,不仅可以使系统开发者在系统建模初期就获得对测试的整体观念,还可以从初始阶段就限制测试例的数量。The Testing and Test Control Notation version(TTCN-3)具有完善、成熟的测试平台,因而将 TTCN-3 作为 U2TP 转换的目标测试语言。U2TP 测试模型到 TTCN-3 Abstract test suite(ATS)自动转换系统的开发,可以在一定程度上减少在测试中代码量的开发。核心部分介绍了 U2TP 到 TTCN-3 自动转换的设计与实现工作。最后,通过一个具体的 U2TP 例子对转换系统进行了验证。

关键词 U2TP,TTCN-3,ATS,自动转换

中图法分类号 TP311.56 文献标识码 A

Research and Implementation of U2TP to TTCN-3 Automatic Transformation

HOU Xiao-yuan WANG Xian-rong LI Hua SHEN Wei-wei

(College of Computer Science, Inner Mongolia University, Hohhot 010020, China)

Abstract Integrating test information by using U2TP in system modeling stage, on the one hand, enables system developers to obtain the overall concept of testing in early system modeling stage, on the other hand, can limit the number of test cases from the initial stage. TTCN-3 has a perfect, mature test platform, therefore, so we make TTCN-3 as the transformation target test language of U2TP. An automatic transformation system that transformed U2TP test model into TTCN-3 ATS can reduce the amount of code in the test development at a certain degree. The core part describes the design and realization work of U2TP to TTCN-3 automatic transformation. The end of this paper validates the transformation system through a concrete example of U2TP, and makes concrete analysis of the generated TTCN-3 ATS.

Keywords U2TP,TTCN-3,ATS, Automatic transformation

1 引言

1.1 研究进展

UML 是一种可视化建模语言,提供了一种对软件系统进行可视化、详述、构造和文档化的工业标准机制。我们可以从 UML 模型中提取测试信息,进行测试例的设计,然而,由于生成的测试信息缺少目的性,测试生成是一个完全的集合,这就导致了极大数量的测试例。2001 年 6 月,OMG 提出 UML 2.0 Testing Profile(U2TP)。U2TP 是基于 UML2.0 的测试建模语言,对系统相关特征和测试模型特征分别进行建模和描述,可以用于从单元测试到系统集成测试的各个级别的测试建模,结合所有主要的对象和组件技术来使用。通常的软件测试方法是,根据 U2TP 建立起来的测试模型,使用特定的测试语言及平台,编写测试代码,进而对软件进行测试。如果设计并实现一个可以将 U2TP 测试模型转换成特定测试语言描述的可执行测试代码的转换系统,可以在一定程度上减少在测试中代码量的开发。那么,选择一个合适的用

于描述测试的语言对日后的测试工作尤为重要。

TTCN-3 是唯一的专门为测试而设计的国际标准化技术。相比于其他测试语言及平台,TTCN-3 具有如下优势:TTCN-3 在测试平台的开发、测试套和适配器的维护以及人员培训上能够节省更多的时间和开销,TTCN-3 具有系统化、自动化的测试方法,在软件开发的早期设计阶段就能够进行测试,TTCN-3 易于适应已有的测试套,并且可以方便地通过标准化接口(TRI/TCI)对已有系统进行部署测试^[1-3]。TTCN-3 无论是测试描述,还是测试执行、测试平台,都有不可比拟的优势,因此我们选择 TTCN-3 作为 U2TP 模型转换的目标语言。

目前,已经有很多研究人员和科研机构对 U2TP 到 TTCN-3 的转换这一问题展开了研究,并且取得了一些成果。在文献^[4-6]中,Stephan Pietsch, Ina Schieferdecker, Justyna Zander 实现了从 U2TP 到 TTCN-3 抽象测试套的转换,转换规则在元模型层次上定义,将源 U2TP 库的元素映射到目标 TTCN-3 库元素。除此之外,还有一些其他的研究成果,其中

本文受国家自然科学基金,面向属性的 CPN 建模及 On the Fly 辅助的测试生成方法研究(61163011)资助。

侯晓媛 女,硕士生,主要研究领域为计算机软件与理论、协议测试与验证,E-mail:houxiaoyuan0816@126.com;王显荣 男,硕士,副教授,主要研究领域为计算机软件与理论、分布式计算与测试工程;李 华 女,博士,教授,主要研究领域为计算机软件与理论、分布式计算与测试工程;沈维维 女,硕士生,主要研究领域为计算机软件与理论、协议测试与验证。

心思想都是通过 U2TP 元素到 TTCN-3 元素的映射来研究从 U2TP 到 TTCN-3 的转换,并通过实例给出了部分测试模块的转换结果,而对于具体的实现过程介绍得却比较少。

1.2 背景知识介绍

U2TP 为黑盒测试提供了开发测试规范和测试模型的概念,定义了用于设计、可视化、说明、分析、构建以及记录测试系统制品的一种语言。它是一种测试建模语言,应用在各种应用领域的测试系统[7]。

U2TP 使用测试特定概念来扩展 UML,如测试组件、判定、缺省等等。这些概念被组合成测试架构、测试数据、测试行为和定时器。作为一个大纲,U2TP 无缝集成到 UML:U2TP 基于 UML 元模型,并且重用 UML 的语法。

U2TP 按照 4 个逻辑概念组来组织,见表 1。测试架构定义测试规范和测试配置的关于结构方面的概念;测试数据定义在测试过程中使用的测试数据的概念;测试行为定义测试过程中关于动态方面的概念;测试定时器为测试过程中的时间量化定义来定义概念。

表 1 U2TP 概念的概述

测试架构	测试行为	测试数据	定时器
仲裁者	缺省	通配符	定时器
被测系统	日志操作	编码规则	时区
测试组件	测试例	数据池	
测试环境	测试日志	数据分区	
	测试目标	数据选择器	
	验证行为		
	判定		

XML 是包含结构信息的文档标记语言。结构信息既包含内容(文字、图片等),还包含那些内容所扮演角色的指示。几乎所有的文档有一定的结构。标记语言是一种识别文档中结构的机制。XML 规范定义了一种添加标记至文档中的标准方式。XML 文档是一种元标记语言,即一种定义标记语言的语言。

TTCN-3 作为国际化标准测试语言,可通过各种通信端口用于所有类型的响应系统测试规范。TTCN-3 应用的典型领域包括协议测试、服务测试、模块测试、基于平台的 CORBA 测试、APIs 等等。TTCN-3 的目的是要用于独立于测试方法、层次和协议的测试套规范[1]。

1.3 本文结构

本文主要研究从 U2TP 测试模型到 TTCN-3 抽象测试套的转换,设计并实现一个自动转换系统,自动地生成 TTCN-3 抽象测试套,可以在系统设计、开发期间进行测试,从而缩短系统的开发周期,降低开发成本。

本文用到的开发环境及工具主要分为两个部分,第一部分为 U2TP 测试模型的开发环境,因为目前没有专门用于 U2TP 测试模型的开发工具,故本文在 Jdk1.6.0d Java 集成开发工具的基础上,使用 eclipse3.7.2 开发环境,另外通过安装楚凡科技开发的 Trufun UML 插件来构建 U2TP 测试模型的开发环境;第二部分为自动转换系统的开发工具,也是在 Jdk1.6.0 基础上,使用 eclipse3.7.2 集成开发环境。

本文第 1 节为引言部分;第 2 节系统分析了转换的关键问题;第 3 节阐述了 U2TP 到 TTCN-3 自动转换的设计与实现;第 4 节介绍实例及分析,最后总结本文。

2 转换的关键问题分析

2.1 U2TP 模型构建

转换系统实现的是 U2TP 测试模型到 TTCN-3 抽象测试套的自动转换,因而本文的首要工作是建立 U2TP 测试模型;有了 U2TP 测试模型的实际例子,才能进行本文的核心工作——U2TP 到 TTCN-3 自动转换的设计与实现。

在本文选用的 U2TP 测试模型的开发环境中,包图、包的结构图用类图来描述;测试环境的组合结构用复合结构图来描述;测试例的行为以及单元测试行为用序列图来描述;测试组件的行为用活动图来描述;测试组件的缺省行为用状态图来描述。

2.2 U2TP 与 TTCN-3 之间的映射关系

U2TP 与 TTCN-3 元素之间的映射关系是本文设计与实现的开始阶段。只有清楚了 U2TP 到 TTCN-3 元素之间的映射关系,转换的设计和实现工作才能很好地进行。

TTCN-3 是用于 U2TP 开发的一个基础。尽管如此,它们在几个方面上还是有区别的。U2TP 目的在于提供 UML 到 TTCN-3 特征选定的扩展,同时限制/忽略其他的 TTCN-3 功能。从 U2TP 到 TTCN-3 的映射是可能的[7]。U2TP 到 TTCN-3 映射的方法包括以下两个主要步骤:

- (1)采用 U2TP 定型和组合,并将它们分配到 TTCN-3 概念。
- (2)定义收集生成 TTCN-3 模块所需信息的过程。

2.3 XML 基本元素

在 U2TP 模型导出的 XML 文档中,uml:Model 元素的孩子结点为 eAnnotations,其中在第二个孩子结点 eAnnotations 中,其孩子结点为 contents,而每个 contents 则代表了一个 U2TP 测试模型图。

由 U2TP 模型图导出的 XML 文件中的主要元素说明如下:

- (1)contents-U2TP 测试模型图孩子结点
 - 结点 1:nodes-U2TP 测试模型图中结点元素
 - 结点 2:connections-U2TP 测试模型图中连接线元素
- (2)contents 属性
 - 属性 1:name-U2TP 测试模型图的名称
 - 属性 2:xmi:type-U2TP 测试模型图的类型
- (3)nodes 属性
 - 属性 1:name-U2TP 测试模型图中结点元素名称
 - 属性 2:xmi:type-U2TP 测试模型图中结点元素类型
- (4)connections 属性
 - 属性 1:name-U2TP 测试模型图中连接线元素名称
 - 属性 2:xmi:type-U2TP 测试模型图中连接线元素类型
 - 属性 3:target-U2TP 测试模型图中连接线元素的目的元素
 - 属性 4:source-U2TP 测试模型图中连接线元素源元素

2.4 U2TP、XML、TTCN-3 三者对应关系的设计

本节就复合结构图的 U2TP、XML、TTCN-3 三者之间的对应关系做介绍(见表 2),而其他 4 种类型图的 U2TP、XML、TTCN-3 三者对应关系就不再赘述。

在复合结构图中,包元素在 TTCN-3 中对应 group,类元素对应 TTCN-3 中的被测或测试组件,类中的接口对应被测或测试组件上的端口。

表 2 复合结构图的 U2TP、XML、TTCN-3 三者对应关系

level	U2TP	XML 标签	XML 属性 xmi:type	XML 属性	TTCN-3
1	diargam	contents	trufun: TCompositeStructureDiagram		
2-1	test context	nodes	trufun: TPackageNode	name	group
2-2	Test Configuration	connections	trufun: TConnectorConnection	name/source/target	function Configuration (connect/map/disconnect/unmap)
3-1	Test Component/SUT	nodes	trufun: TStructureClassNode	name	sut/ type component
3-2	interface	nodes	trufun: TPortNode	name/ parentNode	port

表 2 中将 contents 作为顶层结点。测试环境的复合结构用复合结构图来描述,这其中主要包含了所有测试例和测试环境相关定义的 TTCN-3 模块定义部分(definition part)。第 2 层结点 connections 在 TTCN-3 中表示了测试组件之间、被测系统与测试组件之间进行通信的连接及映射配置,这包括:发送消息、调用操作,以及对调用操作的回复、接收消息、操作调用以及回复。各个测试组件之间的通信、被测系统与测试组件之间的通信都是通过测试组件、被测系统上的端口来进行的,端口表示为 3-2 层、类型为 trufun: TPortNode 的结点。

5 种图的逻辑关系如下,类图转换成 TTCN-3 代表的是语言中的顶层元素 module,因此其他 4 类图转换成的元素必须包含在类图转换成的元素中。复合结构图转换成的 TTCN-3 元素为 group,包含了相应的测试配置、测试组件之间以及被测系统与测试组件之间的初始连接、初始配置参数。序列图映射到 TTCN-3 时为测试例的行为及单元测试行为,包含在 group 中。状态图转换成的 TTCN-3 元素也包含于 group 中,活动图转换成的 TTCN-3 元素包含在 module 中即可。

3 U2TP 到 TTCN-3 自动转换

3.1 转换系统设计

本节将从转换系统整体设计及详细设计两个角度来介绍自动转换系统的设计工作。

3.1.1 转换系统整体设计

为了解决本文中提出的问题,转换系统必须能够实现从 U2TP 测试模型到 TTCN-3 抽象测试套的自动转换。具体来讲,要设计可以选择任意 U2TP 测试模型导出 XML 文件的可视化界面,有了可视化界面,操作起来就会比较方便,通过对可视化界面的选择操作,就能够自动生成 U2TP 测试模型对应的 TTCN-3 抽象测试套的 TTCN-3 文件。图 1 为转换系统的整体设计流程。

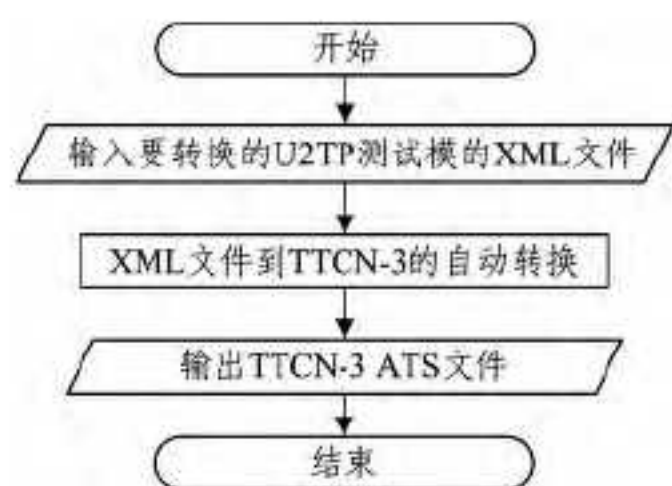


图 1 转换系统的整体设计流程

系统建立一个 Transformation 类,用于处理转换系统的整个逻辑,之后再创建 5 个类,用于处理表示 U2TP 测试模型的 5 类图的 XML 文件,将其转换为对应的 TTCN-3 抽象测试套,分别是 ClassTransformation、CompositeTransformation、SequenceTransformation、ActivityTransformation、StateTransformation。这 5 个类和 Transformation 类之间是泛化的关

系,代表 Transformation 类是一般描述的类,而这 5 个类则作为具体描述的类,即 Transformation 类的子类,如图 2 所示。

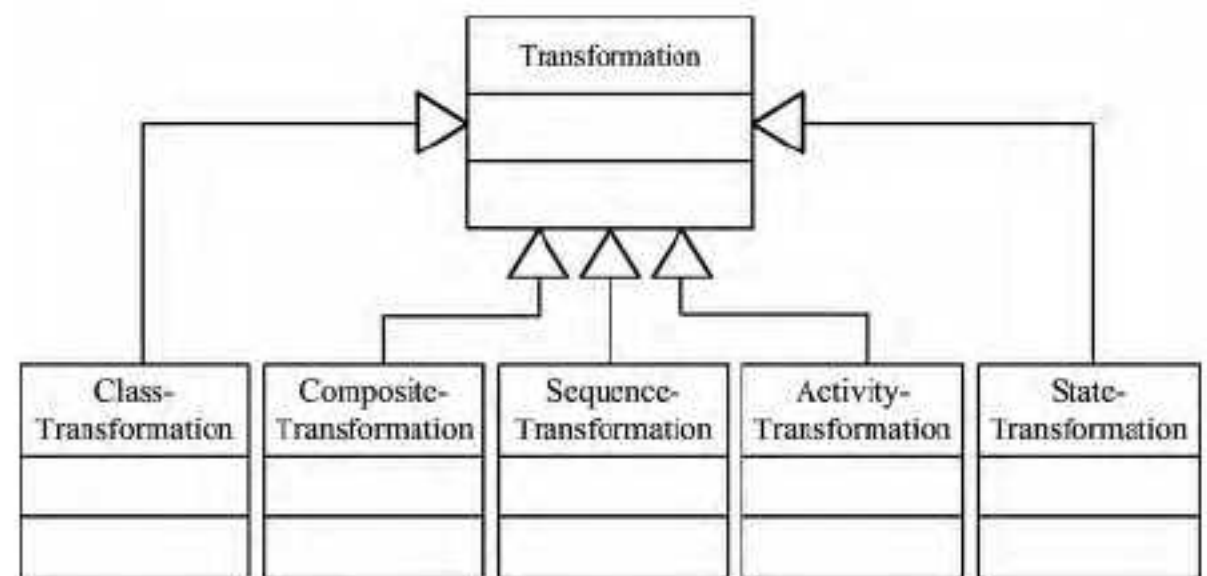


图 2 转换系统类设计

在上述 6 个类的基础上,再创建两个类,分别是 XmlDom 类和 ATS 输出类。XmlDom 类和 ATS 输出类与 Transformation 类是使用(Usage)依赖关系,表示 Transformation 类使用到 XmlDom 类和 ATS 输出类中的属性及操作,如图 3 所示。

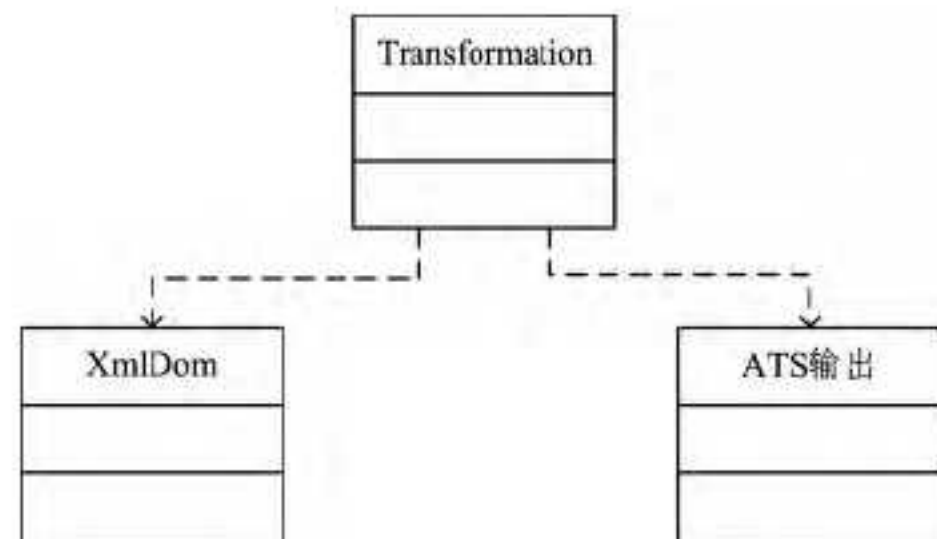


图 3 Transformation 类与 XmlDom 类和 ATS 输出类之间的关系

另外,ClassTransformation 类及 XmlDom 类和 ATS 输出类之间有同样的关系,如图 4 所示。同理,CompositeTransformation、SequenceTransformation、ActivityTransformation、StateTransformation 这 4 个类同 XmlDom 类和 ATS 输出类之间也有类似的关系,在此不一一列举。

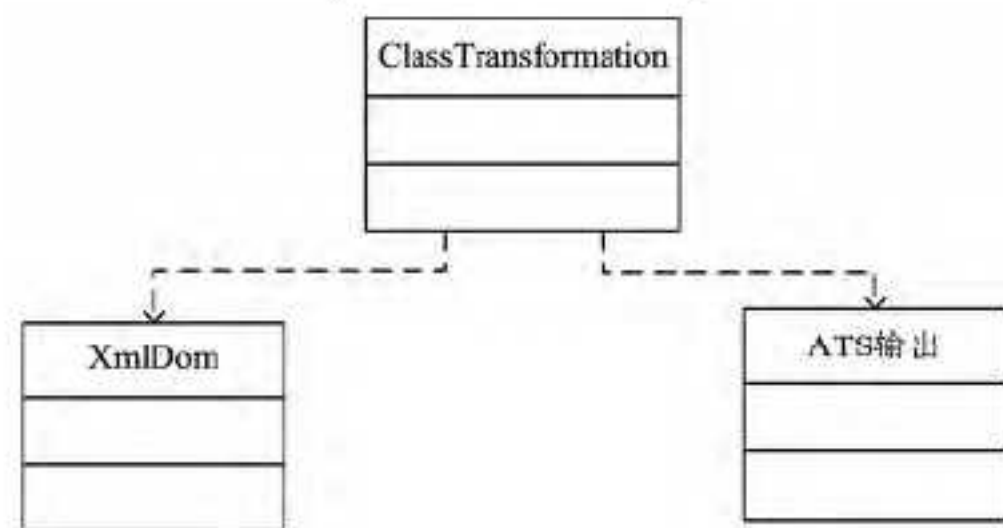


图 4 ClassTransformation 类与 XmlDom 类和 ATS 输出类之间的关系

3.1.2 转换系统详细设计

本节将从转换系统流程设计、类详细设计及算法设计几个方面来对转换系统进行详细设计。

转换系统首先要进行的是将 U2TP 模型转换成的 XML 文档读入到程序中,然后再使用 DOM 对象方法提取相关结点信息,进而通过结点 contents 的类型进行判断,根据结点类型来进入相应的类型图处理程序子模块。

图 5 为 Transformation 类的结构图,其中包含 source-

type、targettype、name 3 个属性及 transform() 方法; source-type、targettype、name 3 个属性分别代表 Transformation 类的源类型、目标类型及名称, transform() 方法表示了从 XML 文件到 TTCN-3 ATS 的转换。

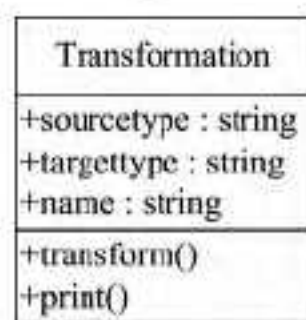


图 5 Transformation 类

图 6 为 ClassTransformation 类的结构图, ClassTransformation 类继承了 Transformation 类, 与 Transformation 类有同样的属性及方法。类似地, CompositeTransformation, SequenceTransformation, ActivityTransformation, StateTransformation 这 4 个类也与 Transformation 类有同样的属性及方法。

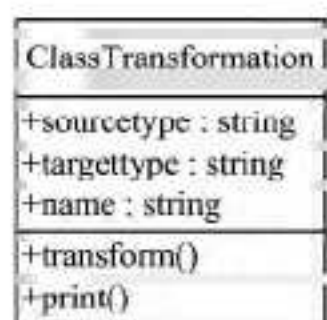


图 6 ClassTransformation 类

图 7 为 ATS 输出类的示意图, 其有 path 和 type 两个属性, 分别表示生成 ATS 目标文件的存放路径及表示类型; ATS 输出类的操作 print() 则代表将转换成的目标 ATS 以文件的形式输出。

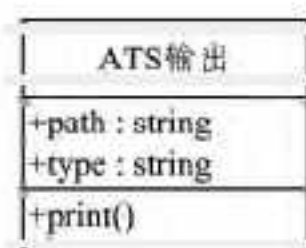


图 7 ATS 输出类

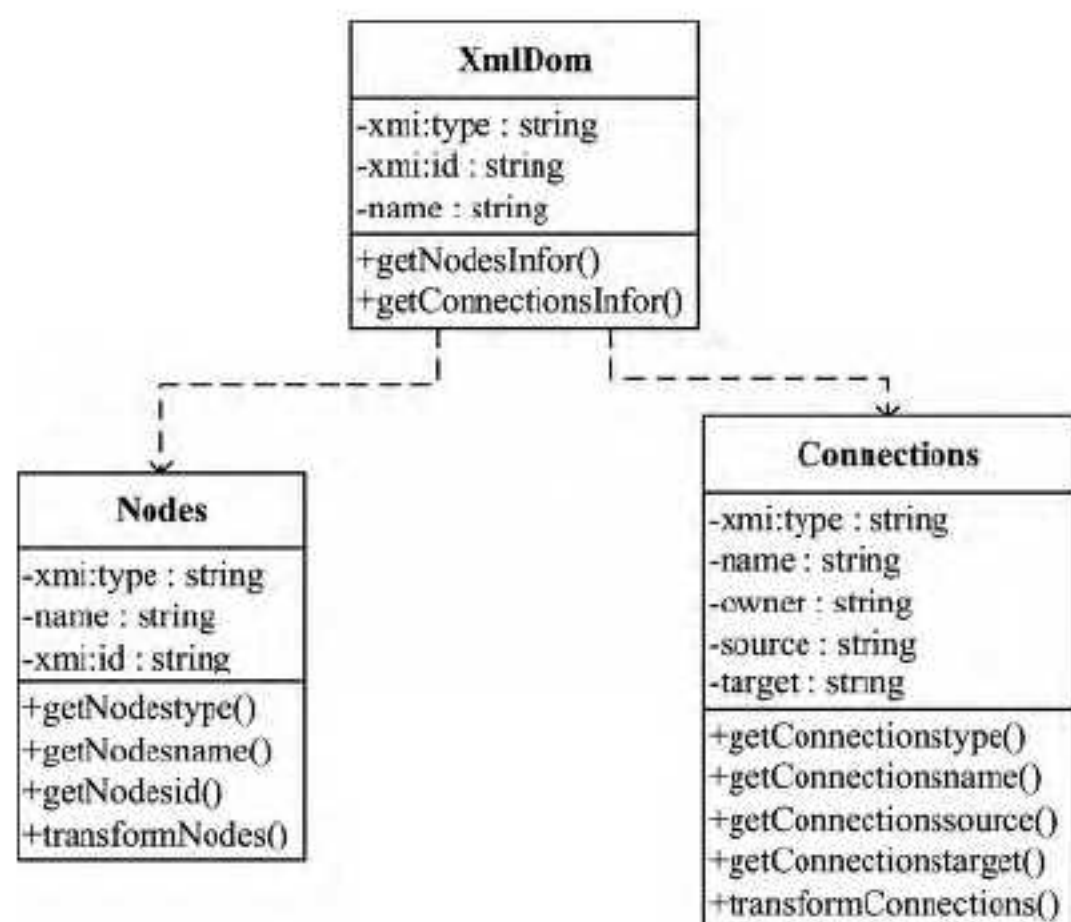


图 8 XmlDom 类与 Nodes 类、Connections 类的关系

图 8 为 XmlDom 类与 Nodes 类、Connections 类的关系, 图中分别显示了 3 个类的属性及操作。XmlDom 类同 Nodes 类、Connections 类之间的使用 (Usage) 依赖关系表明了 XmlDom 类使用到了 Nodes 类、Connections 类中的属性及操作。

转换系统涉及到各个类模块对象之间的交互。图 9 以 ClassTransformation 类为例, 介绍了 Transformation 类的对象与 ClassTransformation 类对象、XmlDom 类对象、Nodes 类对象、Connections 类对象、ATS 输出类对象之间的调用操作。

图中所示均代表各个类的对象。

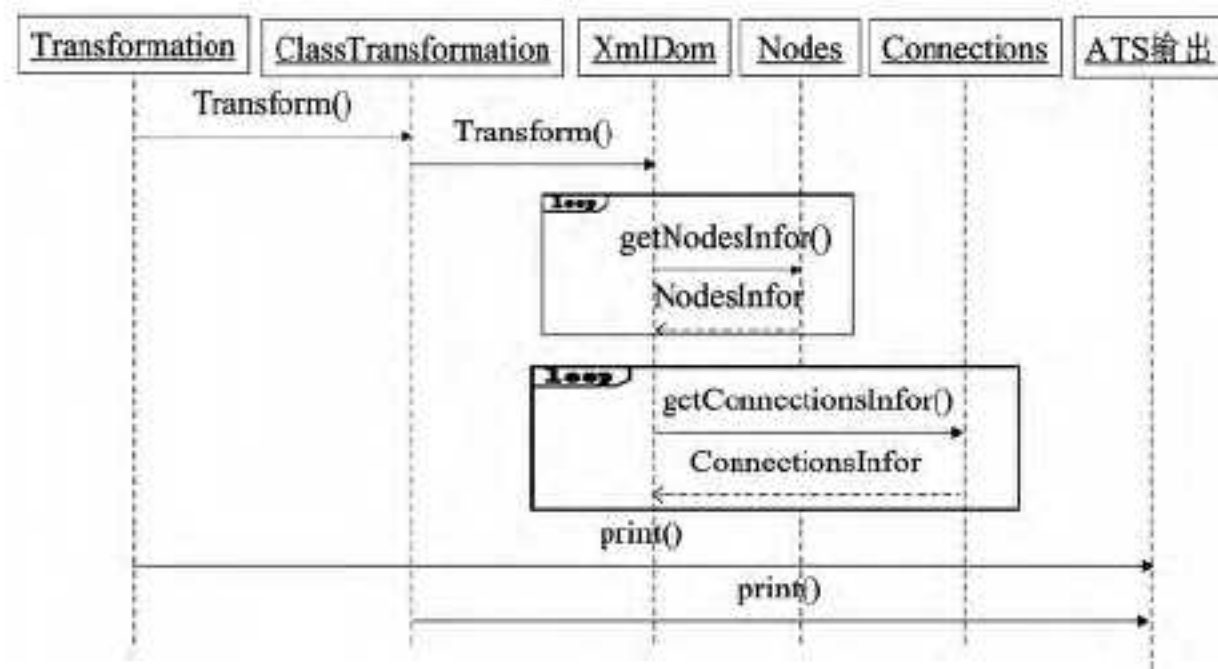


图 9 转换系统中各个类模块之间的交互

算法是系统实现阶段的基础, 下面将以复合结构图自动转换算法为例给出自动转换的算法介绍。

复合结构图自动转换算法如下 (如图 10 所示)。

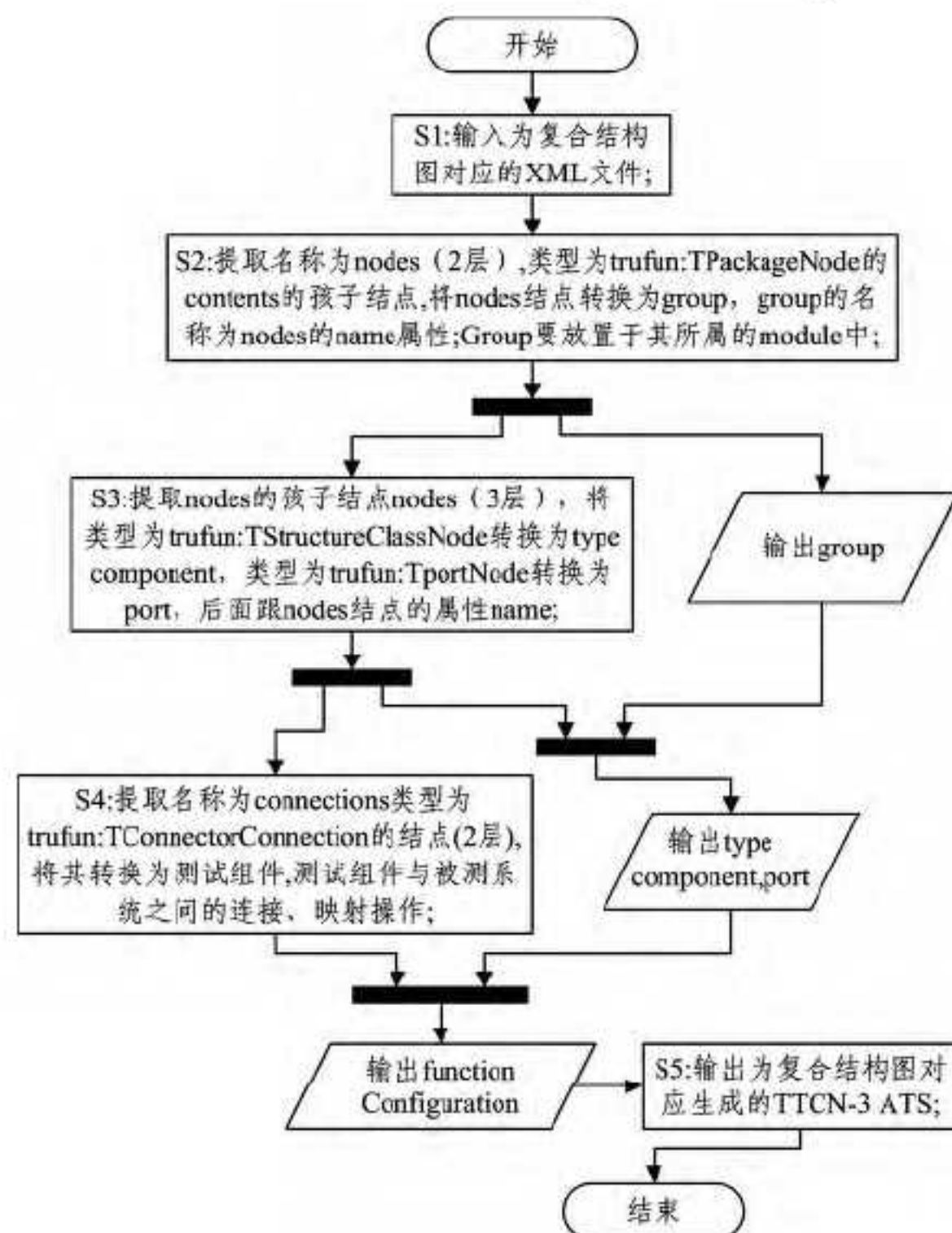


图 10 复合结构图自动转换的算法活动图

S1: 输入为复合结构图对应的 XML 文件。

S2: 提取名称为 nodes (2 层), 类型为 trufun:TPackageNode 的 contents 的孩子结点, 将 nodes 结点转换为 group, group 的名称为 nodes 的 name 属性 (去掉《testContext》关键字); group 要放置于其所属的 module 中。

S3: 提取 nodes 的孩子结点 nodes (3 层), 将类型为 trufun:TStructureClassNode 的结点转换为 type component, 后面跟 nodes 结点的属性 name; 若 nodes (2 层) 的孩子结点 nodes (3 层) 的类型为 trufun:TportNode, 则转换为 port, 后面跟 nodes (3 层) 的 name 属性; 提取其 parentNode 属性, 并将其和所有的 nodes (3 层) 的 owner 属性做比较, 找到与之匹配的, 则将 port 定义归到其定义中。

S4: 提取名称为 connections, 类型为 trufun:TConnectorConnection 的结点 (2 层), 借用数组来存储其 source 和 target 属性, 以此来标识连接的两端结点的 xmi:id 属性, 之后遍历 3 层中的类型为 trufun:TportNode 的 nodes 结点, 对测试组件

之间以及测试组件与被测系统之间的端口进行连接、映射。

S5; 输出为复合结构图对应生成的 TTCN-3 ATS。

3.2 转换系统实现

代码编写是系统实现的首要工作,其主要任务是根据系统设计编写可执行程序代码。本文选用了面向对象编程语言 JAVA 来对转换系统进行实现。

本文对转换系统设计并实现了可视化界面操作,这样可以方便用户使用。本文实现的转换系统能够实现一般的 U2TP 测试模型到 TTCN-3 抽象测试套正确、有效的转换。第 4 节将通过具体的 U2TP 案例来对转换系统进行验证及说明。

4 实例及分析

4.1 U2TP 案例

在转换系统的基础上,我们选用合适的 U2TP 案例进行转换,结果表明,我们的转换系统能够对一般的 U2TP 案例做正确、有效的转换。

本节通过一个具有代表性的 U2TP 小例来进行分析。这个例子通过使用一个银行之间外汇方案来激励。在例中,有一个欧洲联盟银行账户的客户想要存钱到美国的一个自动取款机的账户。

图 11 介绍了 ATM 包的结构, BankATM 控制 ATM 的逻辑,是我们测试的重点。BankATM 类提供了两类端口类型: atmPort 和 bankCom, 其中 atmPort 端口类型的输出方向为 IATM 接口的签名函数,输入方向为 IHardware 接口的签名函数; bankCom 端口类型只有输入方向,为 IBank 接口的签名函数。此外, ATM 包还包含了 OpKind 枚举类型及接口 IATM。

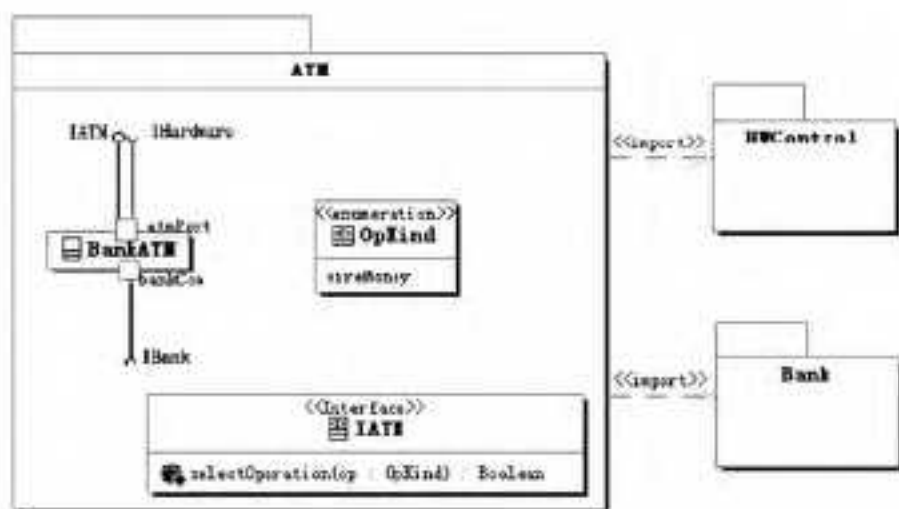


图 11 ATM 包结构图

4.2 目标 ATS 及转换结果分析

本节将给出 U2TP 测试模型经过转换系统的转换生成的 TTCN-3 抽象测试套,并对转换结果做出分析,列出各 U2TP 测试元素与 TTCN-3 元素的对应关系。以下的转换结果与通过手动对 U2TP 测试模型进行转换的结果是一致的,很好地验证了转换系统的实用性。

图 12 中 ATM 包转换结果对应于图 11 中的 ATM 包结构图, ATM 包中 U2TP 和 TTCN-3 元素映射关系如表 3 所列。

```
module ATM {
  import from HWControl all; //导入 HWControl 包的所有内容
  import from Bank all;
  signature selectOperation(OpKind op) return boolean; //签名函数
  type record BankATM {
    type port atmPort-PTYPE procedure {
      in display-> acceptMoney;
      out selectOperation;
    } // 基于过程的端口类型 atmPort-PTYPE 的定义
```

```
type port bankCom-PTYPE procedure {
  in findAccount, wireMoney, checkCredentials
}
} //BankATM 类型定义
type enumerated OpKind {
  wireMoney
} //枚举类型 OpKind 的定义
} // ATM 模块定义
```

图 12 ATM 包转换结果

表 3 ATM 包到 TTCN-3 的映射

U2TP	TTCN-3
package ATM	module ATM
class BankATM	type record BankATM
interface of atmPort	type port atmPort-PTYPE
interface of bankCom	type port bankCom-PTYPE
operation selectOperation	signature selectOperation
class enumerated OpKind	type enumerated OpKind

结束语 本文通过对 U2TP 到 TTCN-3 转换方法的研究,实现了 U2TP 到 TTCN-3 的自动转换。

本文主要做了以下几方面的工作:对 U2TP 概念元素进行分析、整理,并且选取恰当的 U2TP 测试模型开发工具来构建 U2TP 测试模型;分析 U2TP 测试模型的存储 XML 文件,对 XML 标签、属性及其树状结构进行系统的分析;研究 U2TP 与 TTCN-3 概念上的映射关系,并且设计 U2TP、XML、TTCN-3 元素之间的对应关系;对 U2TP 到 TTCN-3 自动转换进行系统分析,通过 UML 描述、主要算法的设计对转换系统做了详细的设计,并且实现了转换;最后,对一个具体的 U2TP 测试实例进行说明,并且通过对其进行转换的结果进行分析,验证了转换系统的实用性及通用性。

本文研究、设计并实现的 U2TP 到 TTCN-3 的自动转换系统,可以对一般的 U2TP 测试模型进行自动转换,具有很强的通用性及实用性,这给软件工程中的测试工作减轻了一定的压力。

参考文献

- [1] ETSI ES 201 873-1 v4. 3. 1: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language. 2011
- [2] ETSI ES 201 873-5 v4. 3. 1: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI). 2011
- [3] ETSI ES 201 873-6 v4. 3. 1: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI). 2011
- [4] Pietsch S, Stanca-Kaposta B. Model-based testing with UTP and TTCN-3 and its application to HL7[J]. Testing Technologies, 2008
- [5] Schieferdecker I, Zhen Rur-dai, Grabowski J, et al. The UML 2. 0 Testing Profile and Its Relation to TTCN-3[M]. Testing of Communicating Systems, Springer Berlin Heidelberg, 2003: 79-94
- [6] Zander J, Zhen Ru-dai, Schieferdecker I, et al. From U2TP Models to Executable Tests with TTCN-3-An Approach to Model Driven Testing[M]. 2005: 289-30
- [7] UML Testing Profile (UTP) Version 1. 1 RTF-Beta 1. 2011