

# 面向 TTCN-3 分布式测试执行平台的 测试适配器框架的研究与设计

王 冠, 吴 际, 徐 珞, 金茂忠  
(北京航空航天大学软件工程研究所, 北京 100083)

**摘 要:** 由于分布式系统具有物理分布、并发访问、时序敏感、平台异构等特点, 所以分布式测试系统必须适应上述特点. TTCN-3 分布式测试执行平台能够满足分布式测试的要求, 但 TTCN-3 测试系统静态绑定测试适配器的特点导致测试系统在单个测试执行节点上不具备同时和多种被测系统交互的能力, 因此本文提出测试适配器框架的概念. 该框架支持 TTCN-3 标准接口规范, 在具备基本测试适配功能的基础上, 整合面向多种被测系统的测试适配器, 并在测试执行过程中根据测试脚本和被测系统的不同动态选择并调用适合的测试适配器, 满足了 TTCN-3 分布式测试执行平台的需求, 同时也增强了 TTCN-3 在分布式测试领域的应用能力.

**关键词:** TTCN-3; 分布式测试; TTCN-3 分布式测试执行平台; 测试适配器; 测试适配器框架  
**中图分类号:** TP311. 5      **文献标识码:** A      **文章编号:** 0372-2112 (2009) 4A-125-06

## Research on Test Adapter Framework for Distributed TTCN-3 Test Execution Platform

WANG Guan, WU Ji, XU Luo, JIN Mao-zhong  
(Software Engineering Institute, Beihang University, Beijing 100083, China)

**Abstract:** Because distributed system had features of physical distribution, simultaneous access, being sensitive to sequence and platform isomerism, distributed test system must fit those features. TTCN-3 distributed test execution platform was able to fulfill requirements of distributes test. However, with the feature of static binding test adapter, TTCN-3 test system can not communicate with different kinds of SUTs at a single test execution node. Therefore the concept of test adapter framework was present in this paper. On the basis of fundamental adaptation with supporting TTCN-3 standard interface, the framework integrated test adapters for different kinds of SUTs, dynamically selected and used adaptive test adapter. As a result, the framework fulfills requirements of TTCN-3 distributed test execution platform, and strengthens the ability of TTCN-3 for distributed test domain as well.

**Key words:** TTCN-3; distributed test; TTCN-3 distributed test execution platform; test adapter; test adapter framework

## 1 引言

随着网络技术的不断发展, 大量分布式系统被部署在网络中, 常见的如 Web 应用、Web Service 等. 分布式系统不同于一般软件系统, 它通常由物理分布的多个子系统组成, 这些子系统通过相互间的协作完成计算任务. 此外, 分布式系统中的诸多子系统在触发方式、工作方式等方面往往也不属于同一类型, 比如, 通过网络协议传递消息的子系统 and 利用本地接口进行方法调用的子系统等. 由于具有物理分布、并发访问、时序敏感、平台异构等特点, 分布式系统测试也不同于一般软件系统的测试.

## 2 研究问题分析

测试分布式系统需要关注以下几个问题: (1) 模拟多个并发用户; (2) 一个模拟用户往往需要和多个不同类型的子系统交互; (3) 在某些应用场景下, 多个并发用户之间需要协同和同步; (4) 为了尽量保持测试环境和系统真实使用环境的一致性, 应该模拟不同用户通过不同物理节点发出测试命令.

通过上述分析可知, 分布式系统的测试系统需要满足以下要求: (1) 支持分布式测试执行, 即在物理分布的测试执行节点上运行测试脚本; (2) 测试系统能够调度、协调各测试执行节点的执行; (3) 由于一个测试用例可

能与多种被测子系统交互, 所以每个测试执行节点应该同时具备和多种被测系统交互的能力. 因此, TTCN-3 分布式测试执行平台 (TTCN-3 Distributed Test Execution Platform, 以下简称执行平台) 成为满足上述需求的解决方案. 执行由测试中心节点 (Test Center Node) 和测试执行节点 (Test Execution Node) 组成. 其中, 中心节点负责管理整个测试活动的执行, 包括脚本部署、调度执行以及结果收集, 而执行节点负责具体测试脚本的执行, 可以和多种被测系统交互.

TTCN-3 (Testing and Test Control Notation 3rd, 测试和测试控制表示法第 3 版) 是一种通用的测试标准规范, 适用于交互式系统和分布式系统的各种黑盒测试<sup>[1]</sup>. 该标准包括 TTCN-3 语言和测试系统架构两部分, 其中语言用于描述测试逻辑, 而测试系统架构则提供了测试系统实现的依据.

图 1 展示了标准 TTCN-3 测试系统的结构<sup>[2]</sup>, 其中系统适配器 (System Adaptor, SA)、平台适配器 (Platform Adaptor, PA) 和编解码器 (Coding/Decoding, CD) 合称 TTCN-3 测试适配器 (Test Adapter, TA), 其作用是适配测试系统与测试系统, 包括通信和数据, 即实现与被测系统的交互. 从作用可以看出, TA 实体是测试系统与测试系统交互的桥梁, 测试系统通过 TA 模拟用户对被测系统的访问行为, 即向被测系统发出激励并接收被测系统的响应. 由此可见, TA 在 TTCN-3 中具有重要作用. 目前, TTCN-3 测试中仍然采用静态绑定的方式来配置相应的 TA, 即无法在测试逻辑层次指定所需的 TA, 导致一旦测试系统实际运行, 就无法动态更改或调整 TA. 因而, 当被测系统有多个不同类型的子系统时, 测试系统就失效了, 因为测试系统在运行中需要动态的调用适合被测子系统的 TA.

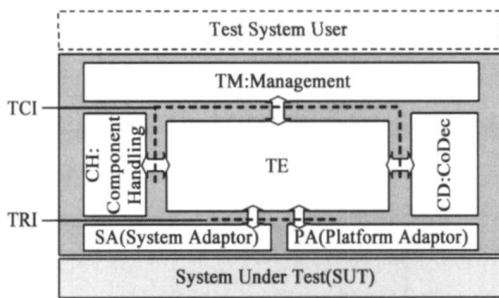


图1 TTCN-3测试系统一般结构图

针对上述问题, 本文提出面向 TTCN-3 分布式测试平台的测试适配器框架 (Test Adapter Framework, TAF). TAF 扩展了 TTCN-3 测试系统的适配层, 在实现基本适配功能的基础上, 其内部根据 TA 与被测系统的对应关系, 在测试执行过程中动态选择并使用与被测系统相适应的 TA. TAF 一方面满足了 TTCN-3 分布式测试执行平台对于一个测试执行节点能够同时与多种被测系统

通信的要求, 同时也提高了 TTCN-3 测试系统自身的灵活性和适用性.

### 3 相关研究

目前 TTCN-3 研究主要集中在两个方面: TTCN-3 应用和 TTCN-3 扩展. 路晓丽<sup>[5]</sup> 和 Schieferdecker<sup>[6]</sup> 分别定义了 IDL 与 TTCN-3 和 XML DTD、XML Schema 与 TTCN-3 间的映射关系, 从而促进了 TTCN-3 在 CORBA 系统和 Web Service 测试中的应用; 高翔<sup>[7]</sup> 在 TTCN-3 语言中添加实时性元素, 提出了 TimedTTCN-3, 扩展了 TTCN-3 语言对实时性的支持. TTCN-3 应用研究是为了扩展 TTCN-3 适用范围, 而 TTCN-3 扩展则是为了增强 TTCN-3 的能力. 与现有研究相比, 本文的研究在上述两个方面都有一定实用价值. 首先, TAF 不是面向某一特定领域或类型的软件测试活动的工具, 而是一种 TTCN-3 测试系统实现方法, 这种方法并不涉及具体的实现平台或者支持平台, 因而具有通用性. 同时, TAF 扩展 TTCN-3 的方法不同于一般对 TTCN-3 语言的扩展, 而是通过采用不同的测试系统实现策略对 TTCN-3 测试系统进行了扩展.

### 4 测试适配器框架的设计

如图 2 所示, TAF 作为 TTCN-3 测试系统的适配层, 首先应该保证具有适配器的基本功能, 因此 TAF 必须满足 TTCN-3 标准接口规范, 即 TRI (TTCN-3 Runtime Interface)<sup>[4]</sup> 和 TCI (TTCN-3 Control Interface)<sup>[3]</sup>, 这两个接口定义了测试系统实体交互的数据以及操作. 这样 TAF 才能与测试系统的其他实体进行正常交互, 同时其内部机制对于其他实体以及被测系统均不可见, 可避免框架内部异常对其他实体的影响. 在 TAF 内部, TAF 并不干预实际适配器的功能, 即是说实际的适配任务依旧是由具体 TA 完成的, TAF 只是将其组织在一起, 并建立其与被测系统之间的对应关系, 因此具体 TA 也必须满足 TRI 和 TCI 接口规范. 而对于 TAF 的内部接口, 则完全独立于 TTCN-3 标准, 这样就将 TAF 对于 TA 的管理机制和具体适配功能分离, 降低二者之间的耦合性.

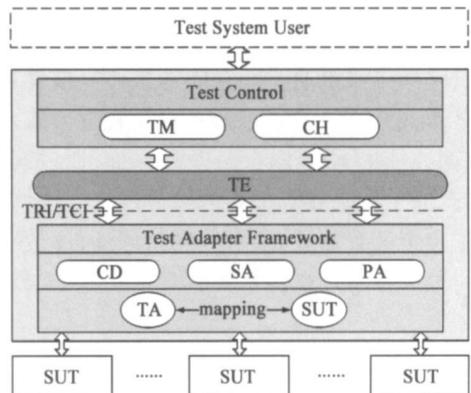


图2 扩展TTCN-3测试系统架构

### 4.1 体系架构

TAF 主要包括 3 个部分: 加载模块、映射模块和调用模块(如图 3 所示). 其中, 加载模块用于将测试系统所处环境中 SA、PA 和 CD 实体加载到内存中, 其输入为环境能力描述和测试适配器属性描述; 映射模块用于建立各类实体与被测系统的映射关系, 其输入为测试适配器属性描述和被测系统属性描述; 调用模块则以实体与被测系统的映射关系为依据, 自动选择并调用适合的实体.

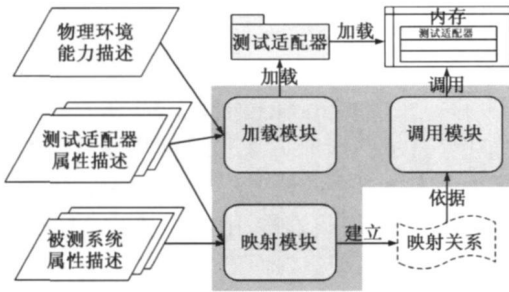


图3 TAF的系统架构

### 4.2 能力与属性定义

TAF 在加载测试适配器和建立测试适配器各实体与被测系统间的映射关系时, 需要描述信息作为依据, 包括环境能力, 测试适配器属性, 以及被测系统属性.

#### 4.2.1 环境能力

测试适配器实体与被测系统之间存在的对应关系, 可以理解为适配器具有支持测试特定系统的能力. 从整个环境的角度看, 当环境中存在特定测试适配器实体时, 实体本身具有的测试支持能力就成为了环境具有的能力. 因此, 环境能力可以用环境中的测试适配器实体表示, 如图 4 所示.

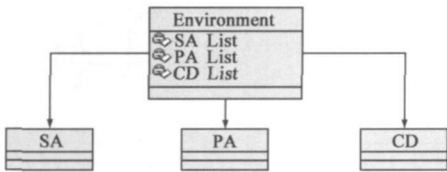


图4 环境能力定义

#### 4.2.2 被测系统属性

在 TTCN-3 核心语言中, 被测系统接收测试系统激励的端口被抽象成为测试系统接口 (Test System Interface, TSI), 参与测试的每一个被测系统端口对应一个 TSI, 因此 TSI 属性可以作为被测系统的属性, 如图 5 所示.

其中, name 代表 TSI 的名称, 是 TSI 的唯一标识, 为

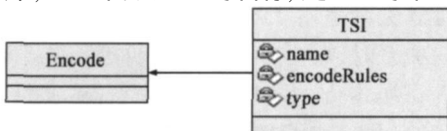


图5 被测系统属性定义

了保证其唯一性, 将其命名规则定为“module 名.TTCN-3 类型名.TTCN-3 实例名”, 其中 module 名是 TSI 所在脚本的 module 名称, TTCN-3 类型名是脚本中定义的 TSI 类型的名称, 而 TTCN-3 实例名则是脚本中声明的该类型实例的名称; encoderules 代表编解码规则, 但不是 CD 实体的编解码规则, 由于通过 TSI 的数据不是单一类型, 所以该属性是所有相关数据类型对应编解码规则的列表; type 代表了 TSI 的类型, 但并非 name 属性中的 TTCN-3 类型, 而是 TSI 的通信类型, 比如网络、本地方法等.

#### 4.2.3 测试适配器属性

TAF 加载 TA, 需要实体加载信息, 包括实体包名、入口类名以及依赖的包名或者类名, 这些信息是 SA、PA 和 CD 三类实体的通用属性. 此外, 因对应被测系统不同, 每个实体都拥有和被测系统相关的独有属性, 这类属性能够表明实体与被测系统之间的对应关系.

(1) 系统适配器 (SA). 测试系统通过 SA 与被测系统通信, 被测系统的通信方式决定了 SA 的功能. SA 支持某种通信方式表明其能够与该类型的被测系统通信, 所以本文定义 port type 属性表明 SA 实体支持的通信方式, 并作为其独有属性.

(2) 平台适配器 (PA). PA 实现外部函数和计时器操作. 外部函数是存在于测试系统之外的功能实现, 计时器操作则包含了控制测试执行时的一些时间机制, PA 实体为测试提供基础服务. 另外, 由于 PA 不是必须的测试适配器, 所以其与被测系统间并没有实质的对应关系. 因此本文将调用外部方法的模块 (module) 名称作为 PA 的属性. 在 TTCN-3 核心语言中, module 是测试脚本的顶层单元.

(3) 编解码器 (CD). 编解码器是测试系统和被测系统数据转换的枢纽, 与测试数据密切相关, 其核心是编解码规则. 编解码规则是从源数据到目标数据的转换规则, 是实现编解码功能的依据. 规则一旦确定, 处理方法和策略也就随之确定. 因此, 本文定义 rule 属性代表 CD 实体依据的编解码规则, 用以表征其编解码特性.

依据上述分析, 本文定义了测试适配器各实体属性模型, 如图 6 所示.

### 4.3 测试适配器与被测系统的映射关系

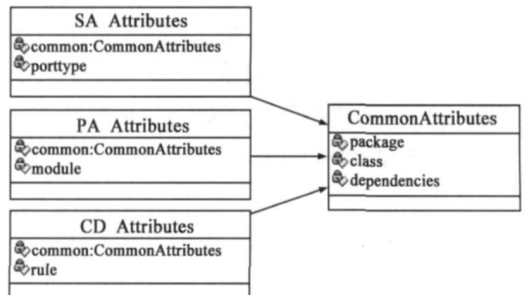


图6 测试适配器各实体属性定义

TAF 的主要功能是根据被测系统的通信特性, 自动选择并调用适合于被测系统的测试适配器, 因此测试适配器与被测系统间的映射关系是 TAF 实现的关键. 映射关系的实质是双方属性间的对应, 但由于每类实体与被测系统之间对应的属性不同, 同时被测系统对每类实体的需求亦不相同, 所以三类实体与被测系统间的映射关系也不相同.

#### 4.3.1 系统适配器(SA)与被测系统

SA 实体的 port type 属性代表 SA 支持的通信方式, 而 TSI 的 type 属性则代表了被测系统具有的通信方式, 二者的一致性 SA 与 TSI 之间映射关系的基础. 一种 SA 只面向唯一类型的被测系统, 而一种被测系统只有唯一类型的 SA 可以支持对其进行测试, 因此二者之间的映射关系如图 7 所示.



图7 SA与TSI对应关系模型

#### 4.3.2 平台适配器(PA)与被测系统

PA 实体为测试提供基础服务, 因此同一个 PA 实体可能在测试不同被测系统时被调用, 而测试一个被测系统也可能同时调用多个 PA 实体, 或者不需要 PA 实体. 由于 TSI 对应唯一的 module, 所以实际中 PA 与被测系统的对应关系如图 8 所示.



图8 PA与TSI对应关系模型

#### 4.3.3 编解码器(CD)与被测系统

CD 实体与 TSI 通过编解码规则关联, 但编解码规则是针对具体数据类型的, 因此 CD 实体与数据类型一一对应. 同时, 由于一类 TSI 至少关联一种数据类型, 所以 CD 实体与 TSI 的对应关系如图 9 所示.

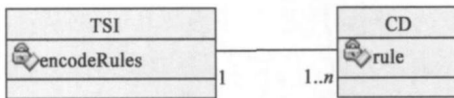


图9 CD与TSI对应关系模型

测试适配器实体与 TSI 间的映射关系体现了本文对 TTCN-3 的扩展, 即利用外部描述信息建立测试适配器与被测系统之间的映射关系, 从而为测试系统指明了面向特定被测系统的测试适配器.

### 4.4 测试适配器框架(TAF)

#### 4.4.1 加载模块

加载模块在测试执行前将环境中所有的测试适配器实体加载到内存中, 环境能力描述和测试适配器属性描述是其输入. 如图 10 所示, 加载模块首先从环境能

力描述获得各实体的属性描述, 再由实体属性描述获得实体的加载信息, 最后利用这些信息将实体加载到内存中.

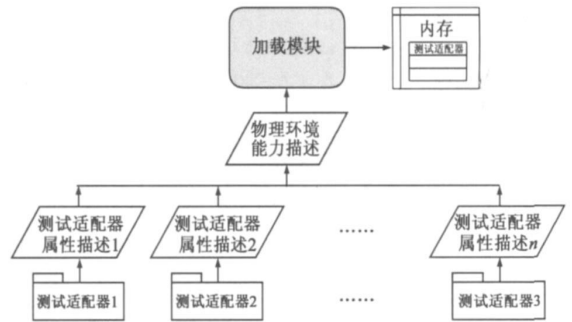


图10 测试适配器加载

描述信息按照级联方式形成两层结构, 避免了在测试适配器实体过多的情况下导致环境能力描述冗长的问题. 而采用全加载策略, 则是为了避免在执行过程中因动态加载失败或错误而出现异常.

加载测试适配器实体是 TAF 整个工作流程的基础, 同时也体现了 TAF 对测试适配器的管理. TAF 根据环境能力描述确定加载的测试适配器集合, 描述信息中项目的增加或减少会使集合发生变化, 加载的内容也将随之改变.

#### 4.4.2 映射模块

映射模块依据实体属性与被测系统属性间的对应性, 分别建立 SA 实体与 TSI, PA 实体与测试脚本, 以及 CD 实体与数据类型三组映射关系, 如图 11 所示.

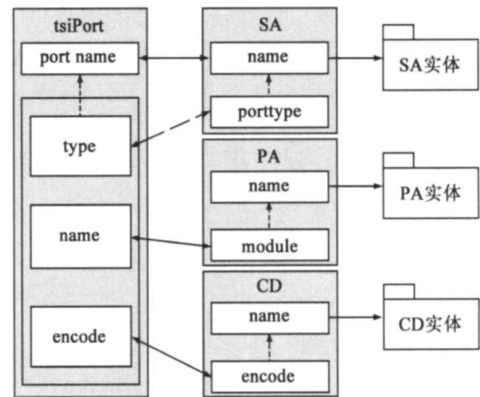


图11 属性间的对应关系

图 11 显示了映射模块建立映射关系的实现机制. 可看出, SA 与 TSI 对应关系的建立不同于其他二者, 主要原因是 porttype 属性与 module 属性、encode 属性在性质上存在差异, 即 porttype 属性是本文定义的, 而 module 和 encode 属性则是由 TTCN-3 核心语言定义的, 所以在建立映射关系时, 需要将本文定义的 TSI 类型属性转化为其在测试脚本中的端口名, 即 portname 属性. 以下算法描述了映射模块分别建立三组映射关系的方法.

算法 1: SA 实体映射关系建立算法

```

INPUT: SAs: = { SAi | SAi is the description to one of SA entities }
SUT: = { TSIi | TSIi is the one of TSIs defined in TTCN-3 scripts }
OUTPUT: Map: = { HashMap < TSI name, SA. name > }
ALGORITHM BEGIN
FOR each TSI in SUT
FOR each SA in SAs
IF( TSI type == SA. porttype)
Map. add( TSI. name, SA. name );
END FOR
END FOR
RETURN Map;
ALGORITHM END

```

### 算法 2: PA 实体映射关系建立算法

```

INPUT: PAs: = { PAi | PAi is the description to one of PA entities }
SUT: = { TSIi | TSIi is the one of TSIs defined in TTCN-3 scripts }
OUTPUT: Map: = { HashMap < TSI module, Vector < PA. name > > }
ALGORITHM BEGIN
FOR each TSI in SUT
module: = TSI. module;
Vector: = NULL;
FOR each PA in PAs
IF( module == PA. module)
IF( Map. notContain( module ) )
Vector. add( PA. name );
END FOR
Map. add( module, Vector );
END FOR
RETURN Map;
ALGORITHM END

```

### 算法 3: CD 实体映射关系建立算法

```

INPUT: CDs: = { CDi | CDi is the description to one of PA entities }
SUT: = { TSIi | TSIi is the one of TSIs defined in TTCN-3 scripts }
OUTPUT: Map: = { HashMap < encode, CD. name > }
ALGORITHM BEGIN
FOR each TSI in SUT
Vector: = TSI. encodeRules
FOR each encode in Vector
FOR each CD in CDs
IF( encode == CD. rule)
IF( Map. notContain( encode ) )
Map. add( encode, CD. name );
END FOR
END FOR
END FOR
RETURN Map;
ALGORITHM END

```

## 4.4.3 调用模块

调用模块是实现自动切换测试适配器的核心组件, 包含两部分功能: (1) 获得被测系统属性; (2) 根据被测系统属性查询映射关系并调用对应测试适配器实

体, 如图 12 所示。

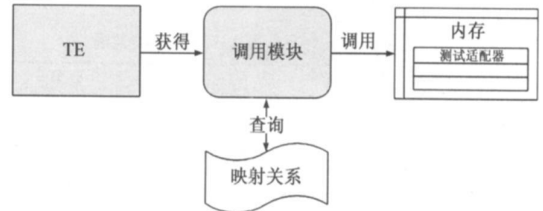


图12 调用模块工作机制

TTCN-3 测试脚本对 TSI 进行了定义, 因此 TSI 的属性, 即被测系统的属性可以从 TE 实体以及相关属性描述中获得。调用模块以 TSI 属性作为条件查询映射关系并获得特定测试适配器的入口信息, 利用入口信息调用测试适配器构成完整测试系统, 从而完成测试适配器的自动切换。

## 5 实验与分析

TAF 可以根据被测系统的不同, 完成测试适配器的自动切换, 使测试系统能够与不同被测系统交互。为了验证 TAF 的设计, 本文选择了文件服务器、UDP 消息服务器和基于 CORBA 的文件审批系统作为测试对象, 利用 TAF 对上述系统进行了测试。

### 5.1 TTCN-3 测试支持工具

为了支持测试, 本文利用了相关 TTCN-3 工具, 包括 TTCN-3 编译器和执行器。编译器用于将 TTCN-3 脚本转换成可执行代码, 而执行器则用于可执行代码的执行。本文采用的是德国 TestingTechnologies<sup>[8]</sup> 公司的 TTCN-3 编译器 TTthree 和执行器 TTman。

### 5.2 测试实现与结果分析

被测系统是分布式在线文件审批系统, 实现了包括提交文件、申请审批、在线审批、文件状态查询以及文件归档等功能。该系统包括三个子系统: 基于 HTTP 协议的文件服务器, 用于提交待审批文件; 基于 UDP 协议的消息服务器, 用于用户与服务器间的简单消息同步; 基于 CORBA 的审批服务器, 用于审批、查询、归档等相关审批操作。实验中测试的流程包括: (1) 用户利用文件服务器提交文件; (2) 用户利用消息服务器通知服务器激活文件审批流程, 并指定审批人; (3) 用户利用审批服务器查询文件状态。此测试流程要求测试系统顺序与文件服务器、消息服务器和审批服务器交互, 因

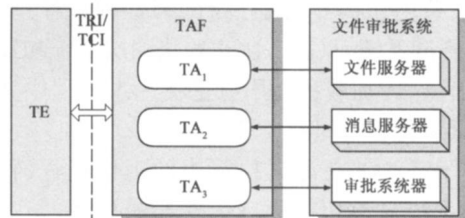


图13 实验测试系统结构

而可以用来验证 TAF 设计的有效性. 图 13 展示了实验中测试系统的结构.

首先, 加载模块会解析环境能力描述文件获取各个实体的描述文件, 并根据各实体的入口信息完成加载. 此实验要对三个被测系统进行测试, 涉及 SA、PA 和 CD 实体共 7 个, 如表 1 所示.

表 1 实体加载列表

类型	包名	类名
SA	FileAdapter.jar	FileAdapter
	UDPAdapter.jar	UDPAdapter
	CORBAAdapter.jar	CORBAAdapter
PA	Pa.jar	Pa
CD	FileCodec.jar	FileCodec
	UDPCodec.jar	UDPCodec
	CORBACodec.jar	CORBACodec

实体加载完成后, 映射模块会根据实体描述和被测系统描述建立测试适配器与被测系统之间的映射关系, 如表 2 所示.

表 2 测试适配器与被测系统映射关系

tsiPort	fileServer	messageServer	corbaSystem
SA	FileAdapter	UDPAdapter	CORBAAdapter
module		CTATestModule	
PA		Pa	
encode	http	udp	corba
CD	FileCodec	UDPCodec	CORBACodec

加载和映射完成后开始执行测试脚本. 在这个过程中, 调用模块会从 TE 实体获得相关属性, 然后根据映射关系调用对应的 SA、PA 或者 CD 实体组成面向每个被测系统的测试系统.

在进行实验的同时, 本文对代码量及执行时间进行了统计, 并与不使用 TAF 的情况进行了比较, 如表 3 所示.

表 3 一般测试系统与扩展测试系统比较

被测系统	一般测试系统		TAF 扩展测试系统	
	代码行数	执行时间	代码行数	执行时间
文件服务器	622		480	
消息服务器	616	22 秒	440	15 秒
审批系统器	2645		2308	

表 3 中代码行数指测试适配器整体代码量. 在 TAF 扩展测试系统中, 由于 TAF 中包含了平台与 TA 之间的关联性代码, TA 只实现适配功能即可, 所以其代码量少于一般测试系统中 TA 的代码量. 在执行方面, 使用一般测试系统测试多个被测系统时需要执行多次测试系统初始化、加载测试脚本、执行测试脚本的过程. 而用 TAF 扩展的测试系统, 只需一次初始化测试系统和加载测试脚本, 而且 TA 加载和调用也是在一次流程中完成的, 因此测试执行总时间减少了.

实验显示了 TAF 在测试执行中的工作流程, 特别是根据被测系统自动切换测试适配器的机制, 与一般方法相比, 本文提出的方法更加灵活, 提高了测试系统

的适用性.

## 6 总结与展望

为了扩展 TTCN-3 测试系统, 使其能够支持不同类型的系统, 本文提出了 TTCN-3 测试适配器框架 (TAF). TAF 由加载模块、映射模块和调用模块组成. 其中加载模块用于加载环境中所有测试适配器实体; 映射模块建立实体与被测系统间的映射关系, 作为选择对应测试适配器实体的依据; 调用模块则根据不同被测系统自动选择并调用相应的实体. 最后本文以 TAF 为核心组件, 通过测试分布式文件审批系统, 验证了 TAF 设计的有效性.

本文已完成的工作实现了既定目标, 但其中仍有可以改进的地方. 例如, 使 TAF 还不能集成已有测试适配器; TAF 采用全加载策略, 当测试适配器较多时, 加载操作会占用可观的资源, 影响测试系统运行效率, 可以考虑改进加载策略优化系统性能.

### 作者简介:



王冠男, 1984 年 4 月生于天津, 在读硕士生, 主要研究方向为软件工程、软件测试.

Email: crown840421@sei.buaa.edu.cn



吴际男, 1974 年 6 月生于安徽肥西, 副教授, 主要研究方向为测试自动化、模型驱动测试、软件性能分析等.

Email: wuji@buaa.edu.cn

徐珞男, 1976 年 1 月生于北京, 在读博士, 工程师, 主要研究方向为软件工程、软件测试.

金茂忠男, 1941 年 8 月生于上海, 教授, 博士生导师, 主要研究方向为软件工程、编译技术.

### 参考文献:

- [1] ITU-T Z. 140146: 2006. TTCN-3 - Testing and Test Control Notation [S]. Institute for Telematics.
- [2] Jens Grabowski, Dieter Hogrefe, György Péthy, Ina Schieferdecker, Anthony Wiles, Colin Willcock. An introduction to the testing and test control notation (TTCN-3) [J]. Computer Networks, 2003, 42(3): 375-403. (下转第 117 页)

- [13] Lowe D G. Distinctive image features from scale-invariant interest points [ J ]. International Journal of Computer Vision, 2004, 60( 2 ): 91- 110.
- [14] Arya S, Mount D M, Netanyahu N S, Silverman R, Wu A. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions [ J ]. Journal of ACM, 1998, 45( 6 ): 891- 923.
- [15] Hartley R, Zisserman A. Multiple View Geometry in Computer Vision [ M ]. Canberra, Australia: Cambridge University Press, ISBN: 0521623049, Jul. 2000.
- [16] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms [ J ]. International Journal of Computer Vision, 2002, 47( 1- 3 ): 7- 42.
- [17] Beauchemin S S, Barron J L. The computation of optical flow [ J ]. ACM Computing Surveys( CSUR ), 1995, 27( 3 ): 433- 466.
- [18] Pérez P, Gangnet M, Blake A. Poisson image editing [ J ]. ACM Transactions on Graphics, 2003, 22( 3 ): 313- 318.
- [19] Szeliski R. Image Alignment and Stitching: A Tutorial [ R ]. Technical Report, Report No. MSR-TR-2004-92, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA, Dec. 2004.

## 作者简介:



刘春晓 男, 1979 年生于山东莱阳, 现为浙江大学 CAD&CG 国家重点实验室博士研究生, 主要研究领域为基于图像和视频的绘制技术、图形图像处理、模式识别与智能系统。  
E-mail: liuchunxiao@cad.zju.edu.cn



彭群生 男, 1947 年生于湖南新化, 浙江大学 CAD&CG 国家重点实验室教授、博士生导师, 主要研究领域为真实感图形、虚拟现实、分子图形学和可视化技术。  
E-mail: peng@cad.zju.edu.cn

杨颖振 男, 1983 年生于湖南新化, 硕士, 主要研究领域为计算机图形学和计算机视觉。

王进 男, 1971 年生于安徽萧县, 博士, 主要研究领域为计算机视觉, 基于图像的建模与绘制技术。

陈为 男, 1976 年生于湖南邵阳, 博士, 副教授, 主要研究领域为科学计算可视化和实时绘制技术。

(上接第 130 页)

- [3] ETSI. European Standard(ES) 201 873-6 V3. 1. 1(2005-06). Methods for Testing and Specification(MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface(TCI). European Telecommunications Standards Institute(ETSI)[S]. Sophia-Antipolis, France, also published as ITU-T Recommendation Z. 140, 2005.
- [4] ETSI. European Standard(ES) 201 873-5 V3. 1. 1(2005-06). Methods for Testing and Specification(MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface(TRI). European Telecommunications Standards Institute(ETSI)[S]. Sophia-Antipolis, France, also published as ITU-T Recommendation Z. 140, 2005.
- [5] 路晓丽. Web 应用软件的测试技术研究[D]. 西北大学, 2006.
- [6] Ina Schieferdecker, Stephan Pietsch, Theofanis Vassiliou-Gioulos. Systematic testing of internet protocols- first experiences in using TTCN-3 for SIP[A]. Proceedings of 5th Africom Conference on Communication Systems[C]. Cape Town, South Africa, May 2001.
- [7] 高翔, 蒋凡. TTCN-3 研究综述[J]. 计算机工程与科学, 2004, 26(16): 17- 20.
- [8] T Tworkbench, TestingTechnologies[OL]. <http://www.testingtech.de/products/ttwb-intro.php>, 2007.