

基于 Eclipse 平台的 TTCN-3 测试系统

尚思超, 赵会群, 陈 萍

(北方工业大学信息工程学院, 北京 100041)

摘要: 分析 TTCN-3 标准规范中的 TTCN-3 测试系统通用结构, 提出一种基于 Eclipse 平台的 TTCN-3 测试系统体系架构。使用 Eclipse 的插件机制, 实现 TTCN-3 测试环境与 Eclipse 平台的集成。讨论测试系统中各模块的功能及相关实现方法, 阐述测试系统的具体测试流程。实际应用表明了该测试系统的有效性。

关键词: TTCN-3 标准; 测试系统; Eclipse 平台; 插件

TTCN-3 Test System Based on Eclipse

SHANG Si-chao, ZHAO Hui-qun, CHEN Ping

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

【Abstract】 This paper analyzes general structure of a TTCN-3 test system in TTCN-3 standards. It proposes architecture of a TTCN-3 test system based on Eclipse. Using Eclipse plug-in mechanism, it implements a TTCN-3 test system based on Eclipse. It discusses the function and implementation of system modules and presents the test process of the test system. Practical application shows the efficiency of the test system.

【Key words】 TTCN-3; test system; Eclipse; plug-in

1 概述

TTCN-3 是由欧洲电信联盟发布的一个国际标准, 是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的测试。其典型的应用领域是协议测试、服务测试、模块测试和基于CORBA平台的测试^[1]。目前, 在为应用 TTCN-3 测试技术而开发的测试工具中, 中国科学技术大学开发的 TTCN3Runner 已经被应用于 SIP(Session Initial Protocol)协议测试^[2]。它在协议测试领域做出了一定的贡献, 但是该软件只适用于 TTCN-3 的开发, 忽略了在 TTCN-3 运行时环境中分别需要实现 TTCN-3 运行时接口(TTCN-3 Runtime Interface, TRI)^[3]与 TTCN-3 控制接口(TTCN-3 Control Interface, TCI)^[4]的被测系统适配器与编解码器的开发, 使得用户无法在同一个开发环境下完成 TTCN-3 的完整测试。

2 TTCN-3 测试系统的通用结构

文献[3]对 TTCN-3 测试系统的概念模型进行了描述。TTCN-3 测试系统的结构如图 1 所示。

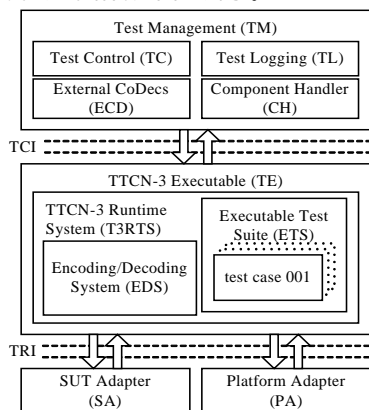


图 1 TTCN-3 测试系统结构

TTCN-3 测试系统由一组具有特定功能的实体组成。这

些实体管理测试执行过程, 解释并执行编译过的 TTCN-3 代码, 实现与被测系统的正确通信, 调用外部函数以及处理定时器的操作等。TTCN-3 测试系统由测试管理 (Test Management, TM)、TTCN-3 可执行实体(TTCN-3 Executable, TE)、被测系统适配器(SUT Adapter, SA)和平台适配器 (Platform Adapter, PA) 4 部分组成。

TTCN-3 测试系统中有 2 个重要的接口: TRI 和 TCI。它们分别指定了 TM 与 TE 之间的接口和 TE 与适配器(SA 和 PA)之间的接口。

TM 负责全面的管理, 包括用户接口的实现和测试执行的管理, 例如测试系统执行前的准备, 最后判决结论的收集以及向 TE 传送模块参数等, 并且 TM 还保存一份测试执行的记录作为测试日志。

TE 负责解释和执行 TTCN-3 抽象测试套。它又可分为 3 个相互作用的实体: 可执行测试套 (Executable Test Suite, ETS)、TTCN-3 运行时系统 (TTCN-3 Runtime System, T3RTS) 和编码/解码系统 (Encoding/Decoding System, EDS)。ETS 解释和执行测试用例, 控制测试顺序以及匹配测试事件。T3RTS 管理 ETS 和 EDS, 并且通过 TRI 和 TCI 两个接口与 TM, SA 和 PA 等实体交互。EDS 负责对测试数据进行编码/解码。ETS 和 EDS 都不直接通过 TRI 和 SA 交互, 而是将返回值传送到 T3RTS, 通过 T3RTS 和 SA 交互。

SA 的目的是适配 TTCN-3 测试系统和被测系统之间所有基于消息的通信和基于过程的通信到一个特定的执行平台。

基金项目: 北京市自然科学基金资助项目“基于模型的测试方法及网络游戏软件测试关键技术研究”(4062012)

作者简介: 尚思超(1983 -), 男, 硕士研究生, 主研方向: 软件测试; 赵会群, 教授; 陈 萍, 硕士研究生

收稿日期: 2007-11-15 **E-mail:** allen34ssc@sohu.com

它负责发送请求到被测系统和通知 TE 收到的任何测试事件。

PA 负责实现外部函数和定时器的操作。虽然定时器实例可以在 TE 中创建，但它们的真正实现却是在 PA 中。

3 系统设计方案

3.1 系统支撑架构

系统使用 Eclipse 平台作为系统支撑平台。Eclipse 平台是一个开放的可扩展的集成开发环境，它可用于创建、集成和部署各类开发工具。Eclipse 的设计思想是一切皆为插件。除了称为平台运行时的微内核外，它的所有功能都以插件形式实现^[5]。本文设计的 TTCN-3 测试系统由一组功能插件构成，通过 Eclipse 的动态查找插件机制，实现与 Eclipse 平台的无缝集成。

3.2 系统体系结构

依据 TTCN-3 测试系统的通用结构，并结合 Eclipse 平台的扩展机制，设计了一种基于 Eclipse 平台的 TTCN-3 测试系统体系结构(如图 2 所示)。

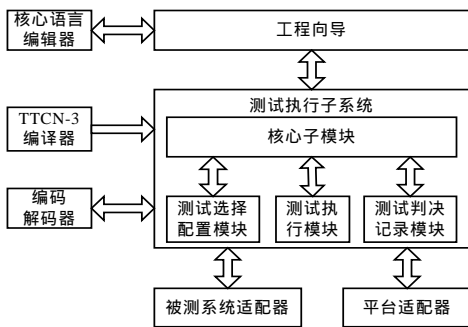


图 2 基于 Eclipse 平台的 TTCN-3 测试系统体系结构

该 TTCN-3 测试系统体系结构具备一定的可扩展性，主要表现在以下 2 个方面。

(1) 开发新插件来扩展系统。随着 TTCN-3 的不断发展，若当前系统无法满足用户要求，可以根据 Eclipse 平台的扩展点和实际需求开发新的功能插件，并有效地插入系统完成系统升级。

(2) 通过系统定义的扩展点扩展原有系统。系统遵照 Eclipse 的自定义扩展点规则定义了一些扩展点，用户可以根据自身需要扩展这些扩展点来增加系统功能。

3.3 系统功能模块

基于 Eclipse 平台的 TTCN-3 测试系统由工程向导、核心语言编辑器、TTCN-3 编译器、被测系统适配器、平台适配器、编码解码器和测试执行子系统等 7 个功能模块组成。

3.3.1 工程向导

工程向导负责新建 TTCN-3 工程，并且用于组织和管理 TTCN-3 资源。工程向导主要利用扩展 org.eclipse.ui.new Wizards 扩展点实现基本向导功能。当通过工程向导创建新的 TTCN-3 工程项目时，系统自动解压 TTCN-3 资源模板包，并且生成 TTCN-3 工程项目目录。由于 Eclipse 平台集成了 Java 开发环境，因此可以在 TTCN-3 工程项目中使用 Java 语言开发相应的被测系统适配器和编码解码器。

3.3.2 核心语言编辑器

核心语言编辑器完全支持 TTCN-3 标准规范，它不仅提供了标准的文本编辑功能，例如打开/保存、复制/粘贴、查找/替换等，还提供了一些对用户友好的附加功能，包括可配置的语法高亮显示、代码折叠、内容助手、文本格式化、内容大纲、代码模板和 TTCN-3 语义检查器等。在对用户友好的

编辑环境下编写 TTCN-3 代码将在很大程度上提高用户的测试效率。核心语言编辑器利用 Eclipse 为扩展编辑器功能提供的 org.eclipse.ui.editors 扩展点和相关 API 实现基本编辑功能与附加功能。其中代码模板功能的设计减少了用户的代码编写量，并且方便用户快速地开发代码框架。它的基本实现思想是根据 TTCN-3 核心语言规范，总结出常用代码模板，通过在代码模板视图与编辑器之间拖拽代码模板标识符，将代码模板插入到 TTCN-3 文本中。TTCN-3 语义检查器可以帮助用户完成 TTCN-3 抽象测试套的静态语义检查，并且输出详细的错误信息。它的实现方法是通过分析 TTCN-3 的语言组成结构，将 TTCN-3 的语言元素按类型划分封装为模型对象(如图 3 所示)。语义检查时，应用 Template 设计模式对 TTCN-3 的关键词进行匹配得到相应的模型对象，当匹配失败时，给出错误代码行数 and 出错原因。

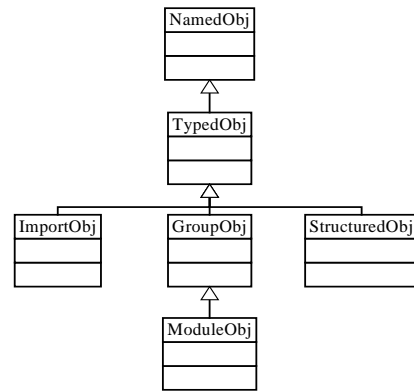


图 3 模型对象类图

3.3.3 TTCN-3 编译器

TTCN-3 编译器使用由德国 Testing Tech 公司开发的 TTthree。TTthree 可将 TTCN-3 抽象测试套编译并打包为同名的 jar 文件，即可执行测试套，并且同时生成模块加载文件。模块加载文件是符合 XML 语法规则的配置文件，文件中包含了被测系统适配器以及测试用例的相关信息。模块加载文件的内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE moduleloader PUBLIC "-//TESTING TECH//DTD
MLF//1.5" "mlf.dtd">
<moduleloader>
  <module File="*.jar" Name="*" Package="generated_ttcn">
    <testadapter Name="*" File="*" />
    <testcase Name="*" Selection="true" Verdict="none"
Status="stopped" Module="*">
      <description />
    </testcase>
    ...
  </module>
</moduleloader>
```

TTthree 编译器为了适应多个 TTCN-3 抽象测试套同时编译的需要，采用了 C/S 结构开发实现，即客户端接收用户提交的参数和 TTCN-3 抽象测试套，然后通过网络传输给服务器端，在服务器端进行编译，并且把编译好的程序打包，再发送回给客户端。TTCN-3 编译器插件对 TTthree 的编译接口进行了封装，并且通过扩展 org.eclipse.ui.editorActions 扩展点进行调用，使得 TTthree 良好地集成到系统中。

3.3.4 被测系统适配器与平台适配器

被测系统适配器负责测试系统与测试系统进行消息通信和过程调用,并且负责发送请求到被测系统和接收响应数据。被测系统适配器需要实现 TTCN-3 运行时接口定义的所有接口,通过分析网络通信客户端与服务器端的交互特点,通常需要实现的 TTCN-3 运行时接口操作的最小集合包括 5 个方法。triExecuteTestCase()负责建立与被测系统的连接。triSend()负责发送测试数据到被测系统。triEnqueueMsg()负责从被测系统接收响应数据,然后经由 TSI(Test System Interface)端口发送到测试套。triMap()和 triUnmap()负责当有多个测试组件映射到一个 TSI 端口时,为从被测系统接收到的测试数据指定正确的目标测试组件。系统中为用户提供了用于实现被测系统适配器的抽象基类,该类中包含了接口的默认实现方法,用户通过继承该类并实现上述最小集合中的 5 个方法完成被测系统适配器的开发。

平台适配器负责实现外部函数和定时器的操作。系统中为用户提供了调用外部函数与进行定时器操作的默认实现方法。

3.3.5 编码解码器

编码解码器负责对 TTCN-3 测试数据进行编码和对接收数据的解码。编码解码器需要实现 TTCN-3 控制接口中定义的 TciCDPProvided 接口。本系统为用户提供了用于实现 TTCN-3 基本类型编码解码的抽象基类,用户需要继承该类并实现 encode()和 decode()方法以完成对 TTCN-3 自定义数据类型的编码解码。

3.3.6 测试执行子系统

测试执行子系统由 4 个子模块组成。测试选择配置子模块负责选择被测系统适配器和编码解码器,配置模块参数以及选择要执行的测试用例。测试执行子模块负责执行可执行测试套和输出测试日志。测试判决记录子模块负责显示测试判决结果、分析测试数据以及导出测试记录。核心子模块向外统一提供控制 TTCN-3 运行环境的接口,并负责各子模块之间的协调。在测试执行时,测试执行子系统需要分别与被测系统适配器、平台适配器和编码解码器进行交互来实现各子模块功能。

4 测试流程

基于 Eclipse 的 TTCN-3 测试系统的具体测试流程(如图 4 所示)按以下步骤执行:(1)使用工程向导创建 TTCN-3 工程项目。(2)在核心语言编辑器中编写 TTCN-3 抽象测试套。(3)通过 TTthree 编译 TTCN-3 抽象测试套,生成可执行测试套和

模块加载文件。(4)在 Java 开发环境中编写被测系统适配器与编码解码器的 Java 源代码。(5)编译 Java 源文件并打包,生成被测系统适配器的 Java 包文件。(6)配置并导入模块加载文件。(7)选择并执行测试用例。(8)显示测试结果并输出测试日志。

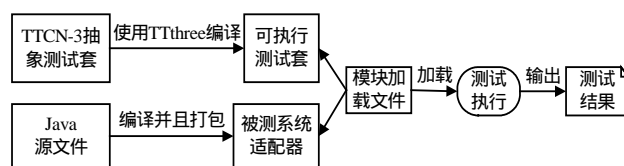


图 4 测试流程

5 结束语

本文提出一个基于 Eclipse 平台的 TTCN-3 测试系统的体系结构,该体系结构具备良好的可扩展性、可维护性和可靠性。所开发的基于 Eclipse 平台的 TTCN-3 测试系统集成了 TTCN-3 开发环境和 TTCN-3 运行环境,加快了 TTCN-3 测试用例的开发、部署和执行,提高了用户的测试效率。该测试系统已经应用于手机网络游戏的测试,并取得了良好的效果。为了使用 TTCN-3 图形表示格式描述 TTCN-3 抽象测试套,下一步将对可视化的 TTCN-3 图形编辑工具进行研究。

参考文献

- [1] European Telecommunications Standards Institute. ETSI ES 201 873-1 V3.1.1-2005 Methods for Testing and Specification (MTS), The Testing and Test Control Notation Version 3(Part 1): TTCN-3 Core Language[S]. 2005-06.
- [2] 蒋凡, 季向东, 曾凡平. TTCN-3 测试系统的设计与实现[J]. 计算机工程, 2005, 31(11): 80-81.
- [3] European Telecommunications Standards Institute. ETSI ES 201 873-5 V3.1.1-2005 Methods for Testing and Specification (MTS), The Testing and Test Control Notation Version 3(Part 5): TTCN-3 Runtime Interface[S]. 2005-06.
- [4] European Telecommunications Standards Institute. ETSI ES 201 873-6 V3.1.1-2005 Methods for Testing and Specification (MTS), The Testing and Test Control Notation Version 3(Part 6): TTCN-3 Control Interface[S]. 2005-06.
- [5] 张云涛, 龚玲. Eclipse 精要与高级开发技术[M]. 北京: 电子工业出版社, 2005.

(上接第 36 页)

参考文献

- [1] 金戈. Linux 高性能集群 - 资源管理和系统管理[EB/OL]. (2002-11-02). <http://grid.tsinghua.edu.cn/home/liulk/publish/computer/ClusterManagement.html>.
- [2] Rajkumar B. 高性能集群计算: 结构与系统[M]. 郑纬民, 译. 北京: 电子工业出版社, 2001: 15-19.
- [3] Pfister G. In Search of Clusters[M]. [S. l.]: Prentice Hall, 1998.
- [4] Wang Choli, Jin Hai, Huang Kai. Designing SSI Clusters with Hierarchical Check Pointing and Single I/O Space[J]. IEEE Concurrency, 1999, 7(1): 60-69.
- [5] Baker M. Cluster Computing White Paper Version 2.0[EB/OL]. (2002-12-02). <http://citeseer.ist.psu.edu/732088.html>.
- [6] 杨冬菊. 单一系统映像 在集群管理系统中的实现[J]. 西北工业大学学报, 2002, 20(2): 292-296.