

基于 Eclipse 的测试工具管理平台构建

赵川^{1,2}, 丁志刚^{1,2}, 宗宇伟¹

(1. 上海计算机软件技术开发中心, 上海 201114; 2. 上海市计算机软件评测重点实验室, 上海 201114)

摘要: 目前市场上还没有在嵌入式软件系统测试领域中进行测试工具和测试用例管理的产品, 该文采用目前方兴未艾的 Eclipse 作为开发工具, 做了一些有益的尝试并取得了一定的成果, 阐述了用 Eclipse 来构建嵌入式软件系统测试管理平台的动机、系统的框架以及技术实现方法。

关键词: Eclipse; 插件; 嵌入式测试; 测试工具; 测试用例

Construction of Test Tools Management Platform Based on Eclipse

ZHAO Chuan^{1,2}, DING Zhigang^{1,2}, ZONG Yuwei¹

(1. Shanghai Development Center of Computer Software Technology, Shanghai 201114;

2. Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201114)

【Abstract】 This paper uses eclipse as development tool and does some work and gets some results in the aspect of managing test tools and test cases in the areas of embedded software test. The reason why it uses eclipse to construct test tools management platform is presented, as well as the platform framework and technical realization.

【Key words】 Eclipse; Plug-in; Embedded test; Test tool; Test case

现有的软件测试管理工具大多偏重于测试过程管理, 也就是对软件的测试生命周期进行管理, 或者对整个软件的生命周期的测试管理, 有的也集成了本公司或者第 3 方公司的测试工具。一般来说, 嵌入式系统的测试离不开测试工具的支持。目前市场上还没有看到在嵌入式软件系统测试领域内集成了第 3 方测试工具的产品。本文在这方面做了一些有益的尝试, 并取得了一定的成果。

由于 Eclipse 和 Java 具有良好的跨平台性和可扩展性, 本文选择了这种开发工具和语言, 并将 Eclipse 作为产品的最终发布形式。Eclipse 是替代 IBM Visual Age for Java 的下一代 IDE 开发环境, 还包括插件开发环境(Plug-in Development Environment, PDE), 这个组件主要针对希望扩展 Eclipse 的软件开发人员, 因为它可以构建与 Eclipse 环境无缝集成的工具^[2]。目前, Eclipse 已经成为大多数程序员的首选, 尤其受到 Java 程序员的欢迎。Eclipse 的版本在不断地更新, 使得定制客户环境和学习都更加简便^[3]。

本测试平台正是利用 Eclipse 的这些特点, 将 Eclipse 作为开发工具, 以开发插件的方式, 使得 Eclipse 也成了产品的最终体现形式, 从而使产品具有良好的可扩展性和可移植性。

1 使用 Eclipse 开发的方法

1.1 Eclipse 的裁减

由于 Eclipse 标准版本本身包含了很多插件, 比如 Java 开发环境, 因此使用 Eclipse 作为产品的最终形式就有必要对它进行一番裁减, 去掉一些对于我们的产品来说并不需要的插件。

Eclipse 的 plug-in 包很多, 在裁减的时候, 主要分为 2 步:

(1) 用编译源代码的方法, 尝试去掉某些包并修改一些 build.xml 文件, 逐步将包缩减, 主要是去掉了 JDT(Java Development Tools)、PDE(Plug-in Development Environment)、Team 等包;

(2) 通过运行的方法, 将包全部去掉并逐个增加, 同时观察运行的 log 文件, 将包继续缩减, 主要是去掉了 debug、doc、sdk 等包和大部分 update 包。这时候 Eclipse 就小多了, 界面也简单了很多。在裁减的同时使用了一些提供观察 Eclipse 启动过程类加载信息功能的插件。另外, 在将开发好的插件发布到裁减后的 Eclipse 中去后, 启动时如果碰到缺少包的问题, 则根据运行 log 文件的提示来逐步增加需要的被裁减掉的包, 直到产品可以正常工作为止。

在项目的开发过程前期, 主要利用对 Eclipse 的裁减来学习和研究 Eclipse, 在实际开发过程中, 并不需要对 Eclipse 进行裁减。在项目后期, 可以利用 Eclipse 的 RCP(Rich Client Platform)^[4], 它是一个建立客户端应用程序所需的 Eclipse 的最小的插件集合。利用这种方法, 可以建立一个比较精简的以 Eclipse 的一部分插件和自己开发的插件组成的应用程序, 并且可以在一定程度上定制自己的界面。

1.2 Eclipse 及其插件的开发

Eclipse 提供了一个 Java 开发工具插件(Java Development Tools, JDT), 这使得它成为一个开发 Java 程序的 IDE 环境。

除了小型的运行时内核之外, Eclipse 中的所有东西都是

基金项目: 上海市科委科技攻关基金资助项目(035115023, 4DZ15009)

作者简介: 赵川(1979—), 男, 硕士、工程师, 主研方向: 软件测试; 丁志刚, 副研究员; 宗宇伟, 高工

收稿日期: 2005-09-21 **E-mail:** dzg@ssc.stn.sh.cn

插件。从这个角度来讲，所有功能部件都是以同等的方式创建的。但是，某些插件比其它插件更重要些。Workbench 和 Workspace 是 Eclipse 平台的两个必备的插件——它们提供了大多数插件使用的扩展点，如图 1 所示。

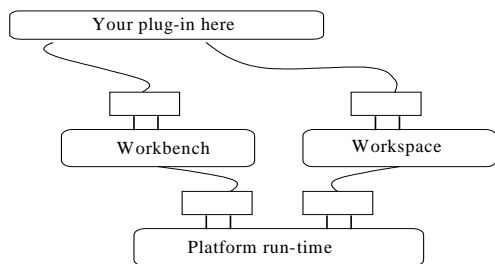


图 1 Eclipse 平台的插件结构

插件需要扩展点才可以插入，这样它才能运行^[5]。使用 JDT 作为开发插件时的 Java 代码的开发环境，使用 PDE 作为开发插件的开发环境。

每个 plug-in 都有一个 plugin.xml 文件，这是 plug-in 的 manifest 文件。在 plugin.xml 中一共有如下 4 个主要的标签：

(1) plugin 标签的属性提供的是插件的基本信息，除了 name、version、provider-name 等，最重要的是 id，它要求不能和现有的 Eclipse 插件 id 有冲突，因此我们用包名作为插件的 id；

(2) requires 标签中所列出的是需要的插件；

(3) runtime 标签指明的是我们开发的插件所在 JAR 包的文件名；

(4) extension 标签是插件扩展点的信息^[2]。

在 Eclipse 中为 plugin.xml 提供了缺省可视化的编辑器，在编写 plugin.xml 过程中可以借助这个编辑器完成一些工作。

2 平台框架

本测试管理平台的框架如图 2 所示。

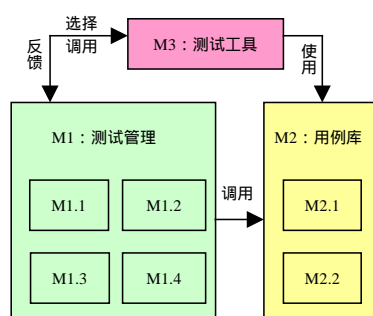


图 2 系统架构

系统各模块名称：

M1：测试管理

M1.1：测试工具登记注册

M1.2：测试工具调度运行定义

M1.3：测试工具启动运行和测试结果获得

M1.4：测试用例库管理

M2：测试用例库

M2.1：功能性测试用例库

M2.2：可靠性(可信性与健壮性)测试用例库

M3：测试工具

3 技术实现

3.1 软件界面

本测试平台主要是在用户界面的活动集(actionSets)和属

性页(preferencePages)扩展点上开发插件。通过添加菜单、工具栏按钮、属性页和向导(Wizard)等方式来创建软件。通过编写 plugin.xml 文件，创建测试管理需要的菜单、工具栏按钮、属性页面等。通过编写相应的类文件，实现相应的功能，并且在属性页面上添加各种控件。还可继承控制台类实现控制台视图用于输出信息。

另外，利用 Eclipse 的 RCP 的开发方式，还对菜单、帮助、窗口的外观等方面进行了定制，使得该软件的界面更加简洁、友好。

3.2 测试工具交互

下面将从测试工具启动运行、监视测试运行、测试结果获得几个方面来讨论技术上的实现。

3.2.1 测试工具启动运行

在本系统的资源配置文件中进行测试工具的设置，包括测试工具的数量、名称、在系统中的存储路径、启动运行参数等。我们使用了 java.lang.Runtime 类的 exec 方法产生一个本地的进程，并返回一个 java.lang.Process 类的实例，本实例可用于控制进程或取得进程的相关信息。Exec 方法中传入测试工具的存储路径加上启动运行参数，作为本方法的参数，执行这个方法就可以启动测试工具并开始运行测试了。

3.2.2 监视测试运行

监视测试运行包括两部分内容：(1)测试运行时测试工具的信息；(2)测试是否结束。

测试工具运行时的信息可以通过 exec 方法获得进程的 getInputStream 方法来得到并同步显示出来。所有的信息全部显示完成后测试也就结束了。这时候可以把测试工具产生的测试报告通过某种方式显示给用户。此外，在某些不便获得测试运行中信息的情况下判断测试是否结束采用了多线程和回调函数的方法来监视测试的运行情况，主要是通过监视测试工具的测试结果文件来判断测试进行的情况，一旦测试结束就会回调主程序的某个方法来显示测试的结果。

3.2.3 获取测试结果

测试结果主要就是测试工具的测试结果。测试工具在测试结束后一般会生成测试结果的报告文件，本系统就是利用这个文件来获得测试的结果。在获取测试结果的时候，由于测试报告文件有时不是基于的文本文件(无法正常显示)，因此先要将它转换成基于文本的文件或者直接利用测试工具查看结果。

3.2.4 应用实例

目前平台中已经集成了 SilkTest、WinRunner、PolySpace、Robustness(Robustness 是项目合作方同济大学开发的一个嵌入式系统健壮性测试软件)、CodeTEST 等软件测试工具。本文以 PolySpace 为例简单介绍平台的使用方法。PolySpace 是一个自动检测嵌入式软件运行错误的软件测试工具，具体操作过程如下：

(1) 点击菜单 Configure->preference 打开配置属性页面，然后在页面左边选择 TestToolsManage，这时页面的右边会出现相关的配置选项。在测试工具名称中选择 PolySpace，设定好工具所在的路径和启动参数，参数中包括待测程序名称和测试结果存放目录等。这些在配置文件中都已经设置好，可以修改相关配置，也可以在属性页面中直接修改。设置好以后点击 OK 按钮保存设定结果；

(2) 点击菜单 TestTools->Start Tool 或者点击工具栏上启动工具按钮，这时会启动 PolySpace 进行测试。在平台的

Console 视图中会出现测试运行的进展情况。当测试完成时,会自动打开 PolySpace 的 Viewer 程序来观察分析测试结果。

3.3 测试用例管理

本系统采用关系数据库来集中存储功能测试用例。测试用例采用自然语言和/或程序片断来描述。因为不同测试工具的工作方式以及测试用例格式并不相同,所以测试工具并不能直接调用测试用例进行测试,而需要进行再次加工。测试用例按照所属领域和产品进行分类。测试平台可以显示出所有的已有的测试用例。可以对领域、产品和测试用例进行增加、删除、修改、搜索等操作。用户在测试某个嵌入式软件系统的时候,在选择测试工具后需要测试用例时,可以根据被测系统所属的领域和产品来查看所有可供参考的测试用例。这时,用户获得的测试用例可能都是用自然语言进行描述的,或者具有简单的程序步骤,用户可以根据这些描述来在测试工具中创建相应的测试用例并在实际测试的过程中使用这些测试用例。

测试用例数据库主要包含 3 张表:

(1)领域表:包含 ID、名称两个字段;

(2)产品表:包含 ID、名称和领域名称 3 个字段;

(3)测试用例表:包含 ID、描述(名称)、前置条件、输入数据、执行动作、期望结果、领域名称、产品名称 8 个字段。

4 结束语

本文简单介绍了嵌入式测试和测试管理软件的状况,阐

述了采用 Eclipse 作为开发工具的动机和好处,给出了系统的架构并描述了具体的技术实现方法。

本测试平台可以运行在 Windows 和 Linux 系统上,支持对第 3 方测试工具、测试用例库的管理。我们在今后的工作中能够增加支持的第 3 方测试工具的数量和种类,改进和测试工具交互的方法,扩充和改进测试用例库。另外,将继续深入研究如何更好地获取第 3 方测试工具对被测系统的测试结果,如何提供可以直接被测试工具调用的测试用例。在本平台的基础上,将研究如何在分布式的环境和架构下建立实时嵌入式软件系统的测试平台。

参考文献

- 1 张君施,张思宇,周承平. 嵌入式软件测试[M]. 北京: 电子工业出版社, 2004.
- 2 倪大鹏. Eclipse 入门: Eclipse 的使用简介及插件开发[EB/OL]. <http://www-128.ibm.com/developerworks/cn/java/l-eclipse/>, 2005.
- 3 谢心. 关注日蚀计划 Eclipse[EB/OL]. http://tech.ccidnet.com/pub/article/c322_a178521_p1.html, 2004.
- 4 Eclipse Foundation. Eclipse Rich Client Platform[EB/OL]. <http://www.eclipse.org/rcp/>, 2005.
- 5 Gallardo D. 开发Eclipse插件[EB/OL]. <http://www-128.ibm.com/developerworks/cn/java/os-eclplug/index.html>, 2005.

(上接第 253 页)

移动通信设备(通过陕西铁通公司的 IP 电话语音网关进行连接)。实验结果如表 3 所示,其中,时延和丢包率为各次实验中音频和视频实验结果的算术平均值,而主观评价分为很好、好、一般、差、很差 5 个等级。另外,当通信对端为固定电话或移动通信设备时,由于无法获得 IP 数据包的 RTT 值,因此没有给出时延和丢包率数据。

表 3 实验结果

对端类型	对端位置	时延 (ms)	丢包率 (%)	音频质量主观评价	视频质量主观评价
UIVP	局域网内	1	0	很好	很好
UIVP	校园网内	2	0	很好	很好
UIVP	兰州大学	25	0~1	很好	很好
UIVP	宁夏大学	20	0~2	很好	很好
UIVP	西安电信宽带	328	0~5	好	一般
UIVP	ADSL 拨号用户	563	0~10	一般	一般
固定电话	西安市	—	—	好	—
固定电话	北京市	—	—	很好	—
固定电话	上海市	—	—	好	—
手机	中国移动	—	—	很好	—
手机	中国联通	—	—	好	—
小灵通	中国电信	—	—	一般	—

由表 3 中数据可见,当对端为教育网内的 UIVP 系统时,时延和丢包率均很低,音频和视频质量的主观评价也很好;而当对端为教育网以外的网络接入用户时,时延和丢包率都有一定上升,音视频质量也有一定程度的下降,这可能是由

于教育网出口的带宽瓶颈造成的;当对端为固定电话和移动通信设备时,由于只有音频通信部分,带宽要求低,因此音频质量优于对端为教育网以外网络接入用户的情况,但是教育网出口带宽的瓶颈依然存在,其音频质量比对端是教育网内 UIVP 的情况要差。

4 结论

本文给出了一种基于 USB 接口的网络可视电话 (UIVP) 的设计,描述了该系统基于区分服务和抖动缓冲区控制的 QoS 保证机制。UIVP 结合了纯软件和独立硬件的 IP 电话系统的优点,具有易用、低成本、节省 IP 地址资源等优点。实际的通信实验证明,在大多数应用环境下,UIVP 系统都保持了较低的时延和丢包率,获得很好的音频和视频质量。

参考文献

- 1 ITU Telecommunication Standardization Sector Study Group 16. ITU-T Recommendation H.323 Packet-based Multimedia Communications Systems[S]. 2000-11.
- 2 IETF Network Working Group. An Architecture for Differentiated Services[Z]. RFC 2475, 1998-12.
- 3 熊光安,张德运. 基于 H.323 的 VoIP 系统 QoS 实现研究[J]. 微电子学与计算机, 2004, 21(4).
- 4 曹玖新,张德运,熊光安. Internet 环境下服务可控的 VoIP 体系结构研究[J]. 西安交通大学学报, 2002, 36(4).