基于 Web 的形式化方法工具 RRTChecker 的研究与设计*)

李 丹[1] 李丹宁[1] Chris George[2]
(贵州科学院 贵阳 550001)[1] (联合国大学国际软件研究所 澳门,P. O Box 3058)[2]

摘 要 形式化方法工具通常是在 UNIX/Linux 系统下设计开发的,难于使用阻碍了形式化方法的进一步推广。本文针对形式化方法 RAISE,提出了一种研究和开发基于 Web 的工具的方法。该方法以原有的 RAISE 工具为基础,通过 Shell 管道拦截技术、ASP 技术、ActiveX DLL 技术及路径重写技术,将工具的所有功能集成整合到统一的、用户友好的 Web 界面上,用户可通过鼠标在浏览器中进行不同的操作。原有的 RAISE 工具的所有功能,在基于浏览器的集成化工具中得到全面支持。该方法也为开发其它形式化方法基于 Web 的工具提供了新思路。

关键词 形式化方法,形式化方法工具,RAISE,Web-based tool,Web 信息系统

Research and Design of a Web-based Formal Methods Tool: RRTChecker

LI Dan[1] LI Dan-ning[1] Chris George[2]
(Guizhou Academy of Sciences, Guiyang 550001, China)[1]
(International Institute for Software Technology, The United Nations University, P. O. Box 3058, Macau)[2]

Abstract Formal methods tools are typically developed and used in UNIX/Linux system, and the difficulties in the use of the tools prevent the formal methods from being more popularized. A method of research and design the Web-based formal methods tool for RAISE is proposed in this paper. Through the adopting of the shell pipeline interception technique, ASP, ActiveX DLL and pathname rewriting technique, the RRTChecker (RAISE Remote Type Checker) is designed as an internet application to support the usage of RAISE tools developed in UNU/IIST through internet. All the functions of the RAISE tools are integrated into a user friendly Web interface, which can be operated by mouse easily. The method proposed in this paper can be employed in the development of the Web-based tools for other formal methods.

Keywords Formal Methods, Formal methods tool, RAISE, Web-based tool, Web information system

1 引言

作为一种以数学逻辑为基础的方法,形式化方法以其严密性越来越受到众多领域的重视,尤其是在安全性和可靠性作为关键问题的系统,如核电站、航空航天、铁路运输系统中得到了较为广泛的应用。但是在工业领域的实际应用中,形式化方法往往被认为太难、太昂贵、太费时,这些可归因于支持工具的缺乏和使用困难。

早期的形式化方法工具,往往是在 UNIX/Linux/Sparc/Solaris 操作系统下开发的,大多也只能在这些操作系统下,用命令行用户界面工作[1]。部分工具开发了 emacs 支持包,在 emacs 环境中使用时,可通过菜单选择不同的功能。迁移到 Windows 操作系统后,很多工具仍然只能通过 DOS 窗口的命令行方式使用。少部分工具,如 B-Toolkit,开发了基于 emacs 或 GUI 的用户界面,但也需要繁琐的下载、安装、学习使用的过程。工具使用的复杂性,限制了形式化方法的使用范围。

"工欲善其事,必先利其器"。Web 技术的发展,为形式化方法工具的研究,提供了新思路。研究基于 Web 的形式化方法工具,提供基于浏览器的、无需安装的、用户友好的使用界面和各种功能的集成,必将极大地降低形式化方法的使用门槛,使实际应用系统的开发人员很容易地使用形式化方法进行用户需求分析、建模等开发工作,提高软件开发的工作效率。

我们的工作将以得到较为广泛应用的形式化方法 RAISE 为目标,研究和开发基于 Web 的形式化方法工具 RRTChecker(RAISE Remote Type Checker)。

2 RAISE 方法及工具

RAISE[2](Rigorous Approach to Industrial Software Engineering,工业软件工程的严格方法)是在一个广谱的规范语言 RSL 的基础上,提供一系列工具和转换技术,形成一种开发软件的严格的形式化方法。RSL 具备强大的描述能力和精确的语义,支持模块化、面向对象、并发控制和实时控制。作为一种广谱语言,RSL 既可用于书写非常抽象的、初级的规范,也可用于书写易于甚至能自动转换到程序语言的更具体的规范。RSL 能够实现逐步求精的开发过程,可以在各个开发层次上使用同一种规范语言,进而处于同一个语义框架内。

为了支持 RAISE 方法,开发了大量的工具。RAISE 最原始的工具由 Terma A/S 公司拥有,只能在 Sun 工作站上运行。1998 年开始在联合国大学国际软件研究所(UNU/IIST) Chris George 的领导下,Gentle 为编译工具,开始开发新系列的 RAISE 工具[3]。首先开发出类型检查器(type checker)。类型检查器(type checker)是一个语义分析程序,它计算定义数据类型的所有语言实体的数据类型属性,并验证这些类型符合语言的类型规则。在其基础上,延伸出一系列的工具:格式打印、模块依赖、可信条件,以及到标准 ML、C++语言和 PVS 的转换器。也开发了从 UML 类图生成 RSL 的工具和

*)贵州省年度计划项目(黔科合(2004)JN057)资助。**李 丹** 助研,研究方向为形式化方法与软件开发技术;**李丹宁** 副研究员,硕士生导师,研究方向为软件技术与数据库安全;**Chris George** 高级研究员,研究方向为形式化方法。

从 RSL 规范生成测试用例[4]的原形工具。目前，新的工具还在不断地开发中。RAISE 工具是在 Linux 下开发的，基本工具只有一个命令行的用户界面，用命令行参数选择不同的功能。迁移到 Windows 操作系统后，仍然只能通过 DOS 窗口的命令行方式使用，使用和调试都很困难。为了改进 RAISE 工具的用户友好性，开发了 emacs 支持包，在 emacs 环境中使用时，可通过菜单选择不同的功能。但 Windows 用户大部分不熟悉 emacs，而且 emacs 包的安装也较为困难，对于改进 RAISE 工具的用户友好性，并无较大帮助。

3 系统架构及功能

RAISE 工具的开发历时近十年，投入了大量的人力物力。基于 Web 的 RAISE 工具 RRTChecker 开发，如果一切从头做起，其工作量是不可接受的。因此只能在原有工具的基础上，进行二次开发。

RAISE 原有的工具使用 Gentle 和 C 开发，除了一些辅助文件外，主要部分经编译后得到一个可执行文件 rsltc. exe，可在 Windows 的 DOS 窗口下以命令行方式执行，通过命令行参数选择不同的功能，结果也显示到 DOS 窗口中。因此，可通过 shell 拦截技术，在服务器后台启动 shell 调用 rsltc. exe，并截获它的输入、输出，并对输出信息进行分析后在更为友好的用户界面上显示出来。整个系统流程示意图如图 1。

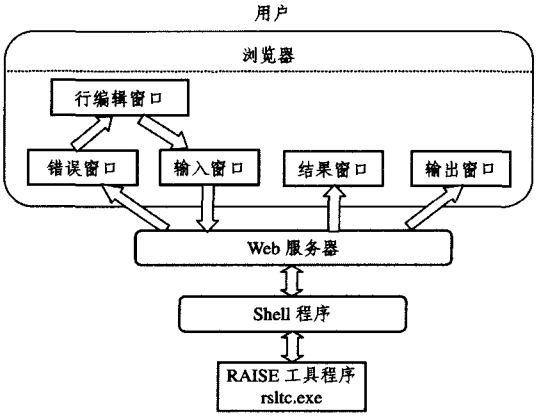

图 1 系统流程示意图

如图 1 所示，RRTChecker 的工作流程如下：

1)用户在浏览器的输入窗口以直接录入、以复制/粘贴的方式录入或读入 RSL 模块文件的方式录入 RSL 程序模块，用鼠标选择不同的功能按钮，将 RSL 模块提交 Web 服务器；

2)Web 服务器负责将 RSL 模块存入服务器上的临时缓冲区，将用户选择的功能构成命令行语句；

3)Web 服务器通过 shell 拦截程序启动 RAISE 工具程序 rsltc. exe，并通过标准输入(stdin)管道将命令行语句传给 rsltc. exe；

4)rsltc. exe 根据命令行语句，对临时缓冲区的 RSL 模块进行相应的操作，结果通过标准输出(stdout)输出。如果选择的是转换到 C++，SML 等功能，rsltc. exe 将在临时缓冲区内生成对应的 C++、SML 程序；

5)shell 拦截程序通过标准输出(stdout)管道拦截输出结果，将其送到 Web 服务器；

6)Web 服务器将得到的输出结果返回到浏览器的结果窗口。Web 服务器同时对输出结果进行分析，如果有错误信息，将生成带错误信息的 RSL 模块(以红色标示错误处)。如果有生成的 C++、SML 等程序，Web 服务器读入后发送到浏览器的输出窗口；

7)如果出现错误，用户点击排错按钮，在错误窗口中打开带错误信息的 RSL 模块，并调出行编辑窗口修改出错处。浏览器将根据行编辑窗口修改输入窗口的对应处；

8)重复以上过程直至得到满意的结果。

整个系统以 Windows 2000 为服务器平台，以微软的 IIS 6.0 为 Web 服务器。客户端为 Windows 操作系统，采用 IE5 或以上浏览器。考虑到开发的方便，系统设计采用了微软的 ASP(Active Server Pages，活动服务器页面)技术。

RRTChecker 系统的主界面如图 2。

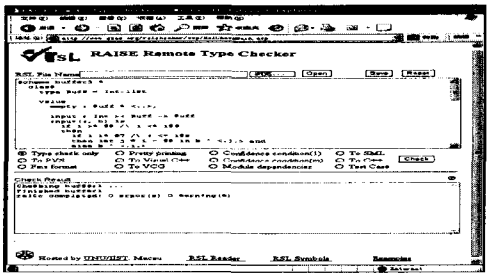

图 2 系统的主界面

RRTChecker 系统集成了下列功能：

- RSL 规范类型检查 (type check)；
- 格式化打印 (Pretty printing)；
- 可信条件 1 和 m 生成(confidence conditions 1 and m)；
- 转换到 SML；
- 转换到 PVS；
- 转换到 C++，Visual C++；
- 转换到 VCG；
- 模块依赖检查(Module dependencies)；
- 生成测试用例(Test Case)；
- 屏幕格式显示(Fine format)。

4 关键技术研究

4.1 shell 程序拦截技术

shell 拦截程序是整个系统运行的核心，负责调用 RAISE 工具包 rsltc. exe，并拦截它的输入输出。

在 Windows 环境下的所谓 shell 程序就是 dos 命令行程序。系统启动 Shell 程序时缺省给定了 3 个 I/O 信道，即标准输入(stdin)、标准输出(stdout)、标准错误输出(stderr)。缺省情况下，系统将管道的输出直接送到屏幕以便直接看到程序运行结果。为了捕获一个标准控制台应用程序的输出，必须把标准输出和标准错误管道输出重定向到自定义的管道。由于 rsltc. exe 只使用了标准输出，在这里只需要重定义标准输出管道。

shell 拦截程序用微软 VB 6.0 编写。用下述语句建立一个匿名管道：

ret = CreatePipe(readPipe, writePipe, sa, 0)

其中 readPipe 用来获取 shell 程序的输出，而 writePipe 可以用来向应用程序发送信息。再把 shell 应用程序的标准输出和标准错误输出都定向到我们预先建好的管道中，再调用建立新进程的函数：ret = CreateProcessA(0&, sCmdline, sa, sa, True, NORMAL_PRIORITY_CLASS, 0&, 0&, start, proc)

(下转封三)

度在3～5s之内，在数据本地存储后，再次登录服务器时，打开同样页面的速度在2～3s范围内，并且只要用户首次打开页面后，就可离线操作，可见运用本地数据存储可提高响应速度，减少用户等待时间。另外运用Flex技术在客户端表现能力方面明显得到提升。

5 并发控制

通过在客户端上存储数据，可以显著改善应用程序的性能和可用性，但必须确保适当地刷新数据并且不会使用陈旧的数据。因为许多用户可以访问和使用相同的数据，必须考虑数据并发的影响。相对于传统Web应用程序优点是：更新可能几乎立即发生，但有时可能发生在数天甚至数周以后。对于可离线操作的应用程序而言，陈旧数据的风险大于始终连接的应用程序。

PRNPAS奖惩考核系统根据实际并发访问的概率不高的情况下，采用了Hibernate的乐观锁[7]解决方案，即为数据增加一个版本标识，通过为数据库表增加一个"version"字段来实现。读取出数据时，将此版本号一同读出，之后更新时，对此版本号序列递增，将提交数据的版本数据与数据库表对应记录的当前版本信息进行比对，如果提交的数据版本号大于数据库表当前版本号，则予以更新，否则认为是过期数据。

结束语 本文在深入研究Flex技术的基础上，针对现有B/S模式应用程序响应速度慢、表现能力差的不足，结合现实用户的需求，提出构建可离线操作的Web应用程序。在此基础上，基于J2EE平台下实现了PRNPAS奖惩考核系统，在实践中得到了较好的应用，实践证明行之有效。在PRNPAS奖惩考核系统中，利用Flex技术中的客户端本地存储，好处在于：

1)可以继续使用现有的应用程序模型(包括J2EE和.NET)，构建更为直观、用户界面更加友好，易于使用、反应更迅速并且可以脱机使用的应用程序。

2)减少用户等待时间，提高系统响应速度，减少带宽成本以及增强客户关系等。

参 考 文 献

[1] Luke W. Web Application Solutions: A Designer's Guide. http://www.lukew.com/resources/WebApplicationSolutions.pdf. 2005,4

[2] Webster S, McLeod A. Developing Rich Clients with Macromedia Flex[M]. Peachpit Press, 2004

[3] Gadge V V. Rich Internet Applications 的技术选项. http://www.ibm.com/developerworks/cn/Web/. 2006,8

[4] 颜金沙，KCLY小土豆工作室. Flash MX 2004 ActionScript 2.0与RIA应用程序. 北京：电子工业出版社，2005(2)：417-425

[5] Bustelo L G, Ruano J. 使用Macromedia Flex开发Web服务客户端. http://www-128.ibm.com/developerworks/cn/webservices/ws-macroflex/. 2004,9

[6] 夏昕，曹晓刚，唐勇. 深入浅出Hibernate[M]. 北京：电子工业出版社，2005(6)：53-77

[7] 李琳骁，王海龙. POJOs IN ACTION中文版：用轻量级框架开发企业应用. 电子工业出版社，2007,4

(上接第287页)

然后，循环读管道里的数据直到无数据可读为止。

整个拦截程序编译为一个ActiveX DLL组件RSLChecker.dll。ActiveX DLL是微软提出的广泛应用于Windows系列的一种代码封装技术，提高了程序代码的可重用性，加快了程序项目的开发速度。

RSLChecker.dll在操作系统中注册后，Web服务器中的ASP程序即可调用这个组件：

```
Set objCheck = Server.CreateObject("RSLChecker.Checkobj")
```

其中，Checkobj是RSLChecker.dll中的类名。

4.2 RSL多模块支持

按照RAISE开发方法[5]，一个用RSL规范描述的系统可以由多个RSL模块组成，这些模块可以分布在多个不同的路径下，模块之间存在复杂的调用关系。这给RRTChecker系统的运行带来了困难，需要调用其它RSL模块时，Web服务器无法到客户端的硬盘上读取。

为了解决这些问题，使用了SessionID标识用户，并使用路径重写技术修改路径信息：

1)对客户端的每一次请求，检查SessionID。如果是新的SessionID，建立一个临时文件目录，作为该SessionID的临时根目录；

2)如果浏览器输入窗口的RSL模块是从文件读入的，检查文件信息，在该SessionID临时根目录中建立相应目录；

3)分析RSL模块中对其它模块的引用信息，将其路径部分影射为临时根目录下的对应目录。

4)将RSL模块存入相应目录。调用shell拦截程序开始工作。

同时，不得不对用户的工作方式作出一些限制，包括：用户RSL模块不能存放在多个硬盘上；必须由最底层模块开始检查，逐步检查到高层模块等。

结束语 基于Web的RAISE工具RRTChecker于2002年在联合国大学国际软件研究所(UNU/IIST)开发完成。经UNU/IIST和贵州科学院数年来的实际应用表明，该系统使用简单，升级容易(换成新版本的rsltc.exe)，特别适用于学习掌握RAISE和开发中小型的RAISE系统。RRTChecker可在Intranet和Internet上使用。该系统在互联网上位于http://www.gzas.org/rslchecker。

以原有的形式化方法工具为基础，通过Web技术，将工具的所有功能集成整合到统一的、用户友好的Web界面上。这种低成本的形式化方法工具开发方法和技术同样可以适用于其它形式化方法，具有广泛的应用前景。

参 考 文 献

[1] Formal Methods Education Resources Tool Pages, Department of Computer Science, Worcester Polytechnic Institute. http://www.cs.indiana.edu/formal-methods-education/Tools

[2] George C, et al. The RAISE Specification Language, CRI A/S, Denmark 1992

[3] George C. The development of the RAISE tools. Lecture Notes in Computer Science, Berlin: Heidelberg, Springer, 2003, 2757

[4] Li Dan, Bernhard K, Aichernig. Combining Algebraic and Model Based Test Case Generation. Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, 2005, 3407

[5] George C, et al. The RAISE Development Method. Prentice Hall, 1995