

Web 应用的自动测试^{*}

周晓宇^{1,2} 许 蕾^{1,2} 徐宝文^{1,2,3} 陈火旺^{3,4}

(东南大学计算机科学与工程系 南京210096)¹ (江苏省软件质量研究所 南京210096)²

(武汉大学软件工程国家重点实验室 武汉430072)³ (国防科技大学计算机学院 长沙410073)⁴

摘 要 Web 应用测试涉及到各种纷繁芜杂的测试对象、方法和过程,需要增强其测试执行过程的自动化、智能化水平以提高测试效率。为此,本文结合 Web 应用的特性,对自动、智能执行 Web 应用测试的必要性、可行性进行分析;在此基础上,详细描述并深入分析 Web 应用测试的执行过程,明确测试执行的步骤、流程以及采用的技术、工具等相关内容;再次,改进捕捉-回放技术使之适应 Web 应用的动态特性,并采用智能 Agent 来实现对整个测试执行过程的监控、管理和应急处理,从而自动、智能地执行 Web 应用测试。

关键词 Web 应用,执行测试,捕捉-回放,智能 Agent

Automatically Testing for the Web Application

ZHOU Xiao-Yu^{1,2} XU Lei^{1,2} XU Bao-Wen^{1,2,3} CHEN Huo-Wang^{3,4}

(Department of Computer Science & Engineering, Southeast University, Nanjing 210096)¹

(Jiangsu Institute of Software Quality, Nanjing 210096)²

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)³

(School of Computers, National University of Defense Technology, Changsha 410073)⁴

Abstract The Web application testing is concerned with numerous and complicated testing objects, methods and processes. In order to improve the testing efficiency, the automatic and intelligent level of the testing execution should be enhanced. So combined with the specialties of the Web applications, the necessity and feasibility of the automatic and intelligent execution of the Web application testing are analysed firstly. On the base of the related work, the executing process of the Web application testing is detailedly described and thoroughly analysed, so as to determine the steps and flows of the testing execution along with the related contents such as the adopted technologies and tools. Furthermore, we improve the capture-replay technology and make it suitable for the dynamic characters of Web applications, and adopt the intelligent Agent to realize the monitor, management and exception-handler of the whole testing execution. Finally, the Web application testing can be implemented automatically and intelligently in this way.

Keywords Web application, Execute testing, Capture-replay, Intelligent agent

1 引言

Web 应用实现了全球范围内的信息共享,其用户数量庞大并增长迅速,因而必须进行系统、全面、充分的测试以保证其质量。但由于 Web 应用具有分布式、动态性、多平台、交互式、超文本等特点且运行环境异构、自治,对其测试的难度较大,要求针对其特性选择测试方法和测试工具。为降低测试成本、提高测试效率,需要一定的辅助工具以提高测试执行的自动化水平;同时要求相应的工具具备一定的智能,以适应 Web 应用动态演化的特性。

Web 应用测试得到了人们越来越多的重视,并取得了一些重要研究成果^[3,6,8,9]。我们在 Web 应用测试的方法、技术和工具等相关方面也进行了一定的尝试^[11~14]。本文主要探讨如何自动化、智能化地执行 Web 应用测试。为此,首先介绍了相关研究工作,分析其必要性和可行性;在此基础上,详细描述

并深入分析 Web 应用测试的执行过程,以明确测试执行的步骤、流程以及采用的技术、工具等相关内容;再次,研究该测试执行过程的具体实现技术,即改进捕捉-回放技术使之适应 Web 应用的动态特性,并采用智能 Agent 来实现对整个测试执行过程的监控、管理和应急处理;最后是结论以及对未来工作的展望。

2 相关研究工作

如果单靠人工执行 Web 应用测试,测试的成本太高,因而可以考虑进行一定程度的自动化测试,以显著提高测试效率。但 Web 应用测试过程中经常会遇到一些变动和异常状况,错误可能发生在服务器端、网络传输过程或者浏览器解析过程等,所给的错误提示信息又非常含糊,给错误的诊断和排除造成很大的困难,因而还需要进一步提高测试的智能化水平。

^{*} 本研究得到国家自然科学基金(60373066)、国家重点基础研究发展规划973资助项目(2002CB312000)、国家自然科学基金青年科学基金(60303024)、武汉大学软件工程国家重点实验室开放基金、江苏省计算机信息处理技术重点实验室(苏州大学)开放基金(KJS02009、T8118011)资助。周晓宇 讲师,在职博士生,主要从事程序设计语言、软件分析及软件测试方面的教学与科研工作。许 蕾 博士生,主要从事软件分析与软件测试等方面的研究工作。徐宝文 教授,博士,博士生导师,主要从事程序设计语言、软件工程、并行与网络软件等方向的教学与科研工作。

由于测试工作通常是按照严格的步骤来执行确定的测试内容,比较机械和单调,适合自动执行,而且测试任务的繁重更能体现自动执行的优势。目前已经有一些测试执行辅助工具应用在性能测试或回归测试中。文[1]详细介绍了捕捉-回放工具(Capture-Replay)的工作过程和相关技术,即首先记录用户在相关页面上所执行的各种动作,然后按一定顺序和内容分类将其存放到合适的位置,最后根据需要查找、匹配相关的执行步骤和动作以进行回放,从而完成自动化的测试。但 Web 应用中有许多动态生成的页面,而且页面中的变动非常频繁,捕捉-回放技术不能完全胜任,因而还需要进一步提高这类工具的智能程度。

智能 Agent 最早出现在人工智能领域,通过一些人工实体来模拟人的行为^[5],具备自治、协作和自学习等良好的行为特征^[7]。因而 Agent 能够自行运行而无需人的指导,使得自己能不断处理新状况并提高自身性能。这些特征使得 Agent 非常适合工作于 Internet 和 WWW 分布式平台,在信息获取、用户个性化服务等方面^[2,4,10]起到了积极的作用。并且 Agent 还能够实现智能化的 Web 应用测试,完成测试用例的自动执行、监控和调度等方面的工作^[11]。

3 Web 应用测试执行方案

执行 Web 应用测试需要全面考虑各方面的影响因素,通常分测试输入、测试执行和测试输出三个组成部分。如图1所示,测试输入部分是预先准备好的测试用例;然后在一定的客户端配置情况下,通过测试监控器组织、管理测试者执行测试任务;完成测试后,将结果输出到下一个步骤,进行分析比较和度量反馈。

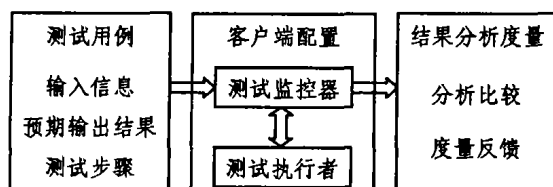


图1 Web 应用测试执行过程

生成选择测试用例是进行 Web 应用测试的先决条件,通常其步骤如下:首先结合测试需求和 Web 应用结构内容得到测试路径,从而得到测试的先后执行顺序以确定测试步骤,并获得待测对象所包含的各种信息以确定输入信息;然后需要确定预期输出结果,通常是结合 Web 应用本身和测试需求进行分析,对于一些动态场景还需要预先多次运行测试用例,归纳运行结果以做下一次测试的参考;在已生成的测试用例的基础上,还需要进一步精简,选择数目适当、质量相当的测试用例进行测试。测试用例的质量直接影响到测试的效果,因而需要花费较多的测试成本。我们在文[13]中提出了一种基于组合测试的测试用例生成方法,并将其应用到 Web 兼容性测试中,能够以较少数目的测试用例覆盖尽可能多的情况。

在测试执行部分中,客户端配置提供了测试执行的环境,具有一定的代表性;监控器是测试执行的核心部分,它直接获取测试用例,负责控制、协调整个测试任务,对突发事件采取一定的处理措施,并且保证执行者之间的负载均衡及协作通信,通常由测试人员或者一些智能实体来担任;执行者是测试任务的主要完成者,按照一定的顺序和步骤执行测试用例,工作繁重而单调,即根据测试用例中规定的测试步骤和输入信

息,依次访问需要测试的站点或页面,完成规定的动作,然后将运行结果输出到下一个阶段,测试用例中所包含的预期输出结果也由测试监控器转交给下一过程,从而可以进行测试结果分析比较和度量反馈。

4 实现技术

为了实现自动化、智能化的 Web 应用测试,需要结合捕捉-回放技术和智能 Agent 进行调整和优化,从而适应 Web 应用交互、动态且变动频繁的特性,以顺利完成测试执行任务。

4.1 获取测试行为

通常采用捕捉-回放技术来实现 Web 应用测试的自动执行,因而首先需要获取测试人员的测试行为。我们在测试端设置一个 Agent,通过 Agent 的记录、学习和分析等活动来捕捉测试人员的执行动作、输入信息等内容,利用 Agent 的自学习能力获取更加精确的测试行为。

图2显示测试端 Agent 由记录器^[7]、分析器和知识库三部分组成。该 Agent 运行在测试执行环境下,其工作过程如下所示:当测试人员执行测试动作时,Agent 中的记录器同时开始工作,依次捕捉、记录所发生的动作及顺序,包括链接的点击情况、提交的窗体内容等;一段时间以后,分析器对记录器中的内容进行处理,根据测试任务的不同对测试执行动作分类,以此获得测试者针对各类测试任务所采用的相应测试动作,并将其存放在知识库中。上述过程还可以根据实际情况的变化而不断更新,从而确保知识库中内容的精确性和时效性。

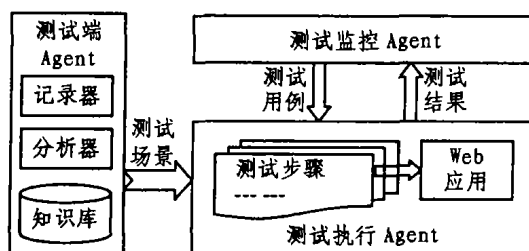


图2 测试执行 Agent 工作过程图

由于 Web 应用测试任务繁多,不可能捕捉到所有的测试场景;动态生成的页面和频繁发生的变动使得捕捉到的测试场景不能直接使用,甚至不能使用。但通过 Agent 获取的测试行为具备更好的可操作性,因为经过 Agent 的学习、分析,能够根据测试任务的不同给出对应的测试过程,具备广泛的适应性,同时测试过程非常详尽,在细节上也给出了明确的指导。

4.2 自动执行测试

通过智能 Agent 捕捉到测试行为以后,同样可以通过 Agent 来自动回放测试动作,即选择具备一定智能的 Agent 作为测试执行者,替代测试人员完成测试任务。测试端 Agent 和作为测试执行者的 Agent 分别完成捕捉-回放任务,在内部构成和功能实现上都有较大的区别。

如图2所示,测试执行 Agent 在测试监控器的控制下执行测试任务,即:首先根据测试任务的类别,从测试端 Agent 的知识库中选择对应的测试场景;然后测试监控器提供测试用例给测试执行 Agent,并结合测试场景提供相应的输入数据,从而得到确定的测试步骤和执行动作;测试执行 Agent 依次访问需要测试的 Web 应用页面,找到动作发生点,完成事先记录的动作,然后将测试结果输出给测试监控器,再由其

输出到下一个阶段。

测试执行 Agent 在回放测试动作时,由于 Web 应用中存在着动态生成的页面以及发生了变动的页面,使得预先记录的动作不能直接使用,需要采用一定的方法来解决这一问题。通过引入模糊匹配和智能识别技术,可以使得测试执行 Agent 更好地适应 Web 应用特性,即对由提交同一个窗体而动态生成的不同页面,由于采用了同样的模板而具有结构上的相似,因而通过模糊匹配技术可以迅速发现内容上的差异;通过对页面元素的智能识别,该 Agent 能准确发现待测页面上的变动之处,进而调整相应的测试方案。

4.3 监控测试过程

测试监控器负责控制、协调整个测试任务。为了提高测试的智能化程度,将测试人员彻底解放出来,并便于与测试执行 Agent 的通信和交互,我们也采用智能 Agent 作为测试监控器,从而具备表述、联系上的一致性。结合图1、图2可以看出测试监控器是整个测试执行过程的枢纽所在,关联着测试的输入和输出,并且需要时刻监控各个执行 Agent 的运行情况。监控 Agent 的工作过程大致可描述如下:当测试用例作为输入信息发送给监控 Agent 以后,该 Agent 首先对测试任务的规模做出准确的判断,以决定所需要的测试执行 Agent 的数目,并将测试任务分割成一个个单元发送给各个测试执行 Agent;然后监控 Agent 根据各测试执行 Agent 的运行情况实时进行调度,保证各执行 Agent 的负载均衡以及相互之间的协作通信;当测试过程中出现一些异常状况时,监控 Agent 采取应对措施,及时排除故障、消除影响。

上述功能的实现需要事先对监控 Agent 进行大量的训练,以形成一定的规律和准则,并将其存放到监控 Agent 自身的知识库中;同时,利用 Agent 所具备的自学习特性,不断提升智能程度,直至能够独立自主地解决一些突发事件。

结束语 Web 应用测试执行过程通常需要花费大量的测试成本,并且测试效率不高。为此需要引入一些测试辅助工具,以提高测试执行过程的自动化水平;同时要求相应的工具具备一定的智能,以适应 Web 应用动态演化的特性。在已有工作的基础上,本文建立起一个高效的 Web 应用测试执行方案,即改进捕捉-回放技术并应用智能 Agent 技术,使之适应于 Web 应用的动态特性,以获得精确的测试行为,然后通过智能匹配实现正确回放,并有效实现对整个测试执行过程的监控、管理和应急处理,其实现技术具备一定的自动化、智能化水平。

未来的工作主要集中在 Web 应用测试执行过程实现技术的研究上。由于测试过程分成前期准备、测试执行和后期比

较三个阶段,相互之间关系密切、彼此影响,因而首先需要分别对各阶段的实现细节展开深入研究,然后再将其集成为一个有机联系的整体,从而进一步优化和完善该测试执行过程,使之具有更好的针对性和适应性,并研究和开发相应工具。

参考文献

- 1 Anupam V, Freire J, Kumar B, Lieuwen D. Automating Web Navigation with the WebVCR. In: Proc. of WWW, 2000. 503~517
- 2 Belkin N J, Croft W B. Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM, 1992, 35(10): 29~38
- 3 Gao J, Chen C, Toyoshima Y, Leung D. Engineering on the Internet for Global Software Production. IEEE Computer, 1999, 32(5): 38~47
- 4 Guttman R H, Moukas A G, Maes P. Agent-mediated Electronic Commerce: A Survey. Knowledge Engineering Review, 1998, 13(2): 143~152
- 5 Hewitt C. Viewing Control Structures as Patterns of Passing Messages. Artificial Intelligence, 1977, 8(3): 323~364
- 6 Kallepalli C, Tian J. Measuring and Modeling Usage and Reliability for Statistical Web Testing. IEEE Trans Software Engineering, 2001, 27(11): 1023~1036
- 7 Krulwich B. Automating the Internet: Agents as User Surrogates. IEEE Internet Computing, 1997, 1(4): 34~38
- 8 Kung D C, Liu C H, Hsia P. An Object-Oriented Web Test Model for Testing Web Applications. In: Proc. of the Asia-Pacific Conference on Quality Software (APAQS), 2000. 111~121
- 9 Ricca F, Tonella P. Web Site Analysis: Structure and Evolution. In: Proc. of the Int. Conf. on Software Maintenance (ICSM), 2000. 76~86
- 10 Wooldridge J, Jennings N R. Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 1995, 10(2): 115~152
- 11 Xu Lei, Xu Baowen, Chen Huowang, et al. Test Web Applications Based on Agent. Journal of Software, 2003, 14(Suppl.): 9~16
- 12 Xu Lei, Xu Baowen, Chen Zhenqiang, Chen Huowang. Website Evolution Based on Statistic Data. In: Proc. of the 9th IEEE Int. Workshop on Future Trends of Distributed Computing Systems (FTDCS), 2003. 301~306
- 13 Xu Lei, Xu Baowen, Nie Changhai, et al. A Browser Compatibility Testing Method Based on Combinatorial Testing. In: Proc. of the Int. Conf. on Web Engineering (ICWE), 2003. 310~313
- 14 许蕾, 徐宝文. 用户行为获取方法在 Web 性能测试中的应用研究. 软件学报, 2003, 14(增刊): 115~120