

基于用户行为和会话的 Web 应用测试方法

武晋南, 高建华

(上海师范大学计算机科学与工程系, 上海 200234)

摘要: Web 应用程序难以创建有效的测试用例, 使其既能满足测试的需求, 又能测试 Web 应用程序的多用户交互行为。针对上述问题, 采用从 Web 日志数据中得到域数据创建测试用例的方法, 从 Web 日志中获取用户行为数据和用户会话数据。提出一种基于用户行为和用户会话的测试新方法, 通过实验验证该方法在功能覆盖和故障检测方面的有效性。

关键词: Web 日志; 域数据; 用户行为; 软件测试

Web Application Test Method Based on User Behavior and Session

WU Jin-nan, GAO Jian-hua

(Department of Computer Science and Engineering, Shanghai Normal University, Shanghai 200234)

【Abstract】 It is difficult for Web applications to create effective test cases which satisfy test requirements and can test multi-user alternation behavior. Aiming at the problems, this paper proposes an approach which gets field data from Web logs and uses field data to create test cases, gathers user behavior data and user session data from Web logs and proposes a new test approach for Web application based on user behavior and user session approach. It confirms the approach's effectiveness in coverage and fault detection through experiment.

【Key words】 Web log; field data; user behavior; software test

1 概述

随着 Internet 的广泛应用, 政府网站、商业网站、科技网站等各种网站逐步走进人们的生活。Web 应用程序已成为软件系统中发展最快的一类, 因此, 确保 Web 应用程序的质量和可靠性成为软件测试的一项重要任务。不同于传统软件, Web 应用程序存在控制流和数据流复杂、用户引起事件的不可预期及频繁的更新, 这些都使 Web 测试更复杂, 使有效测试用例的设计变得困难。一种针对 Web 应用程序的功能测试方法是利用捕获和重现机制记录用户引起的事件, 收集并将它们转换为脚本, 重现它们用于测试。记录的事件是发送给 Web 服务器的请求, 包括基本请求和参数-值(name-value)对。对于 Web 应用程序, 基本请求是请求的一种类型, 是没有用户提交相关数据的资源定位, 如 GET/apps/bookstore/login.jsp。本文将基本请求和参数-值对称为域数据。Web 服务器具有记录这些请求的能力。

不同于人为地创建测试用例^[1-3]或创建静态模型^[4-6], 利用 Web 应用程序的域数据创建测试用例是比较有效的方法。目前已有使用域数据进行测试的方法, 如基于用户会话的测试方法^[6], 但该方法忽略了 Web 应用程序的多用户交互行为。本文从 Web 日志数据中得到测试所需的域数据, 结合用户行为, 提出了基于用户行为和用户会话的 Web 应用测试新方法。这种方法减少了人为生成测试用例的花费, 同时又考虑了用户的行为, 满足了 Web 应用程序功能测试的需求。

2 背景及相关知识

2.1 Web 应用原理

Web 应用是一系列 Web 页面和部件形成的系统, 它的执行使用 Web 服务器、网络、HTTP 以及浏览器。用户通过浏览器可以影响系统的状态。一个 Web 页面可以是静态的, 其内容对所有用户都相同, 也可以是动态的, 其内容由用户的

输入决定。Web 应用是多种技术的集合, 包括定义良好的分层体系结构、动态的页面生成技术以及扩展的应用框架。大规模的基于 Web 的软件系统需要成千上万行代码, 包含对象间的多种关联, 其中涉及与用户有意义的关联。另外, 改变用户的操作剖面 and 频繁的维护等都使自动测试变得复杂。

2.2 Web 服务器日志

Web 服务器在响应用户的请求并将用户请求的文件发送出去时, 把请求写入日志。所以, Web 服务器日志记录了用户访问本站点的信息。目前常用的 Web 服务器有 Apache, Netscape Enterprise Server, MS IIS 等, 互联网上最常用的是 Apache。

对于 Apache 的访问日志, 支持多种日志文件格式最常见的是 common 和 combined 2 种模式, 在 httpd.conf 中进行配置, common 为缺省模式, 形如:

```
LogFormat "%h %l %u %t \"%r\" %>s %b"
```

其对应的日志记录为

```
216.35.116.91- [19/Aug/2000:14:47:37-0400]"GET/apache_pb.gif HTTP/1.0" 200 654
```

第 1 项信息(%h)是发送请求到服务器的客户的 IP 地址, 即 216.35.116.91。

第 2 项信息(%l)是用于记录浏览者的标识, 由客户端 identified 进程判断 RFC1413 身份。输出中的符号“-”表示此处信息无效。

第 3 项信息(%u)用于在记录浏览者进行身份验证时提供的名字。输出中的符号“-”表示此处没有要求用户进行身份验证。

基金项目: 国家自然科学基金资助项目(60673067)

作者简介: 武晋南(1984 -), 女, 硕士研究生, 主研方向: 软件可靠性设计; 高建华, 教授、博士

收稿日期: 2009-09-25 **E-mail:** jnwu_2007@126.com

第 4 项信息(%t)是请求的时间。格式为：[日/月/年:时:分:秒 时区]，即[19/Aug/2000:14:47:37-0400]。

第 5 项信息(\\ "%r")是客户端发出的包含许多有用信息的请求行。格式为“请求方式/请求资源 协议版本”。即"GET /apache_pb.gif HTTP/1.0"。

第 6 项信息(%>s)是服务器返回给客户端的状态码。它指示了请求的结果：被成功响应(以 2 开头)；被重定向(以 3 开头)；出错(以 4 开头)；产生了服务器端错误(以 5 开头)。输出中为 200，表示服务器成功响应。

第 7 项信息(%b)是返回给客户端的总字节数，即 654。

另一种访问日志格式是 combined 模式，它与 common 模式的格式类似，但是多了 Referer 和 User-agent 这 2 项，其格式如下：

LogFormat "%h %l %u %t \\ "%r" %>s %b \\ "%{Referer}i" \\ "%{User-agent}i" " combined

对应的日志记录为

216.35.116.91 - - [19/Aug/2000:14:47:37 -0400] "GET /apache_pb.gif HTTP/1.0" 200 654
"http://www.example.com/start.html" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

前 7 项信息与 common 模式的格式相同。

第 8 项信息(\\ "%{Referer}i")指明了该请求是从哪个网页提交过来的。输出中的"http://www.example.com/start.html"应该包含/apache_pb.gif 或其链接。

第 9 项信息(\\ "%{User-agent}i")是客户端提供的浏览器识别信息，即"Mozilla/4.0(compatible; MSIE 6.0; Windows 98)"。

本文通过分析 Web 服务器的访问日志，从中找出用户行为及用户会话，进而生成测试用例，对 Web 应用程序进行测试。Web 服务器 common 模式访问日志的记录示例如下：

116.233.23.161 - - [01/May/2008:00:00:00 +0800] "GET /sd_gg.html HTTP/1.1" 200 1797
116.232.156.132 - - [02/May/2008:00:00:45 +0800] "GET /css/default.css HTTP/1.1" 404 1000
192.168.45.64 - - [01/May/2008:00:00:09 +0800] "GET /footer.html HTTP/1.1" 304 -
219.228.46.34 - - [02/May/2008:00:01:00 +0800] "GET /bookstore/Books.jsp?category_id=3 HTTP/1.1" 200 1023
219.228.46.34 - - [02/May/2008:00:01:05 +0800] "GET /book store/BookDetail.jsp?category_id=3&item=6 HTTP/1.1" 200 1553

2.3 用户行为

对于 Web 应用程序，用户行为主要是浏览网页，通过单击具有超链接的文字和图片，搜索信息并查看结果，提交用户数据，通过表单单击按钮。总体来说，用户行为可概括为 2 大类：

(1)浏览、搜索后的查看等单控制行为。这些行为的特点是用户不需要提交数据。因此，如果用户请求中的域数据结构只包括基本请求，则可判定该用户行为是单控制行为，如 2.2 节示例中的前 3 条记录。

(2)搜索信息、提交信息等复杂控制行为。这些行为的特点是用户需要提交数据。因此，如果用户请求中的域数据结构包括基本请求和参数-值对，则可判定该用户行为是复杂控制行为，如 2.2 节示例中的最后 2 条记录。

2.4 基于用户会话的测试

在基于用户会话的测试中，每个用户会话就是一系列包含基本请求和参数-值对的用户请求。当用户访问应用程序时会被记录下来。例如，示例中最后 1 条记录中的用户请求 URL: BookDetail.jsp?category_id=3&item=6，基本请求是 BookDetail.jsp，参数-值对是 category_id=3 和 item=6。基本请求

可能是访问静态或动态页面内容的 HTTP 请求。基本请求和参数-值对称为域数据。从服务器日志中可以生成基于用户会话的测试用例。一个测试用例是与一系列用户会话关联的 HTTP 请求的集合。

通过对 Web 日志进行分析，标识提取出用户会话；再对收集到的用户会话使用不同的策略生成测试用例^[4-5]。文献[6]提出了 3 种生成测试用例的方法：(1)US-1 是直接重用来自用户会话的数据；(2)US-2 是组合不同用户会话数据；(3)US-3 是混合常规的用户请求和可能有问题的请求(如在提交表单的同时进行导航操作)。本文利用 Web 日志中的域数据，结合用户行为，提出了基于用户行为和用户会话的新方法(UBS)。

3 基于用户行为和用户会话的测试

为了简单，本文定义当一个请求来自于一个新的 IP 地址时作为用户会话的开始，当用户离开 Web 站点或会话超时则认为用户会话结束。

$U=\{u_1, u_2, \dots, u_m\}$ 表示一个用户会话集，任意的 $u_i \in U$ 包含 n 个请求 $r_{i1} \sim r_{in}$ ， n 为任意数，并且任意的 r_{ij} 包含 $url[game-value]^*$ 。“*”表示零或多次发生。

$B=\{B1, B2\}$ 表示用户行为的集合，其中， $B_i=\{b_{i1}, b_{i2}, \dots, b_{in}\}$ ， b_{in} 是表示用户行为的域数据并且是唯一的， n 为任意数； $B1$ 是单控制行为的集合，任意的 $b_{1i} \in B1$ ， b_{1i} 仅包含基本请求 URL； $B2$ 是复杂控制行为的集合，任意的 $b_{2i} \in B2$ ， b_{2i} 包含 $url[name-value]^*$ 。

(1)基于单控制行为和用户会话的测试方法 UBS-1 包含以下 8 个步骤：

- 1)选择单用户行为集合 $B1$ ；
- 2)从 U 中选择一个未使用的会话 u_i ；
- 3)生成一个新的测试用例，将 $B1$ 中用户行为的域数据复制到测试用例 T_i 中；
- 4)选取 $r_{ij} \in u_i$ ；
- 5)确认 T 中是否包含 r_{ij} ，若包含，则跳转到 4)；若不包含，则将 r_{ij} 添加到测试用例 T_i 中；
- 6)将 r_{ij} 标识为“已确认”，从 4)开始重复执行，直到所有的用户请求都已确认，将该用户会话标识为“已确认”；
- 7)跳转到 2)重复执行，直到所有的用户会话都已确认；
- 8)对生成测试用例进行测试。

(2)基于复杂行为和用户会话的测试方法 UBS-2 与 UBS-1 方法生成测试用例的步骤基本相同，只是将单用户行为集合 $B1$ 改为复杂用户行为集合 $B2$ 。

4 实验

本文通过实验来验证基于用户会话和用户行为的测试方法在覆盖率方面的有效性。

4.1 变量及度量标准

表 1 描述了在实验中考虑的生成测试用例的 Web 测试方法。使用覆盖率为度量标准。覆盖率是指功能和基本模块在生成 page 页面及访问数据库的代码中所占的百分比。

表 1 Web 测试方法

标签	描述	类型
US-1	直接重用用户会话	User Session
UBS-1	基于单控制行为和用户会话的测试方法	User Behavior & Session
UBS-2	基于复杂行为和用户会话的测试方法	User Behavior & Session

4.2 电子商务网站

本文选择了一个免费、开源的在线购书电子商务网站，可以在 gotocode.com 得到。对于网站的功能，主要关注顾客的行为。这是因为收集的用户数据与顾客的活动行为有关。顾客可以使用这个网站搜索、浏览、注册、管理购物车、在线订购。这些功能是通过 ASP.net(C#)脚本及一些用于处理数据和动态生成 HTML 页面的模块实现的。在代码部分用于提供顾客功能的主要有 67 个功能。

4.3 日志数据处理

为了获得用户行为和用户会话数据，在校园里招募了志愿者，在一段指定的时间内，让他们扮演顾客的角色，鼓励他们按照自己的习惯使用这个电子商务网站。在该 Web 应用程序的服务器上可以得到的原始日志数据。

为了便于运用测试用例生成方法，需要对日志数据进行处理，去除日志中不相关和无效的记录，通常有 3 种情况：

(1)一般情况下，用户不会显示请求站点中的图形文件和页面样式文件，这些文件通常是站点根据请求页面中的连接自动下载的，所以，URL 中以 .jsp, .jpeg, .JPG, .JSPG, .gif, .GIF, .css 和 .js 等结尾的记录都可以删除。

(2)用户请求访问失败的记录，这类访问的返回代码为 4xx(请求错误)或 5xx(服务器错误)等。

(3)用户请求方法中不是 GET 的记录也可以删除。

日志数据进行预处理之后，只选取对测试方法有用的字段、IP 地址以及访问请求，然后对数据进行用户行为和用户会话的识别。

4.4 故障插入

本文试图从故障检测方面评估 Web 测试技术的性能。然而，该网站的应用程序中不存在故障，为了获得故障，找几个计算机专业的学生，他们至少有 2 年的编程经验。要求他们根据自己的经验插入故障。故障的类型包括以下 3 种：

(1)脚本故障。这种故障与变量有关，如定义、删除或值的改变；还有与控制流相关，如添加新的块、重新定义执行条件、移除块、改变执行次序、添加或移除功能调用。

(2)表单故障。这种故障包括添加、删除或修改一个表单的名称或一个名称的预定义值。这种故障可以种植在动态生成 HTML 的脚本部分。

(3)数据库查询故障。这种故障包括修改查询表达式。修改可能影响操作的类型、访问的表、表中的字段、检索关键字或记录的值。

在该购书电子商务网站中，一共插入了 41 个故障。

4.5 实验框架

本文构建了一个实验框架，如图 1 所示，包括 4 个步骤：(1)从 Web 服务器上获得日志数据。(2)对日志数据进行预处理，标识用户会话及用户行为。(3)测试用例的生成。使用基于用户会话的方法(US-1)以及基于用户会话和用户行为的方法(UBS-1,UBS-2)。(4)测试方法的有效性评估，主要从功能覆盖和故障检测两方面进行评估。

测试集的执行方法是，将每个测试用例自动转化为一连串的 HTTP 请求，并送往相应的 Web 应用。为了度量测试技术的功能覆盖率，需要应用工具，使用另外的脚本代码片段。功能每执行一次，其相应的计数器就会增加。为了评估测试集的错误检测有效性，单个地激活每一个故障，并执行每一个测试用例，比较相关的输出结果，以确定测试用例是否揭示了故障。

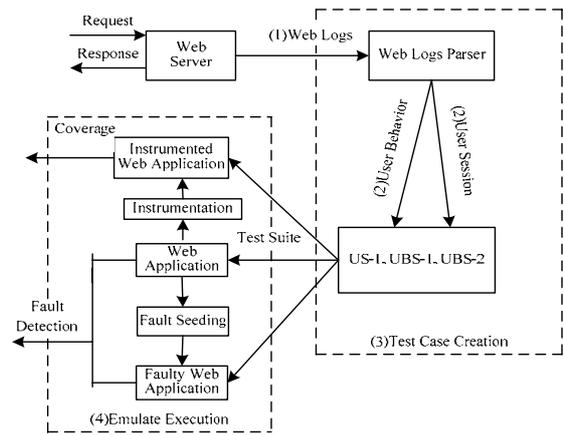


图 1 实验支持框架

4.6 实验结果

表 2 显示了 Web 测试方法在功能覆盖率和故障检测方面的有效性。从表 2 中的数据可知，基于用户行为和用户会话的测试方法(UBS)优于基于用户会话的方法，并且基于单控制行为和用户会话的测试方法(UBS-1)比基于复杂控制行为和用户会话的测试方法(UBS-2)更有效。

表 2 覆盖率和故障检测有效性 (%)

度量标准	功能覆盖率	故障检测率
US-1	97	56
UBS-1	98	61
UBS-2	97	60

5 结束语

本文研究了 Web 测试的新方法，通过对 Web 日志的分析，得到用户行为数据和用户会话数据，提出了基于用户行为和用户会话的测试方法，并通过实验验证了该方法在功能覆盖和故障检测方面的有效性。

使用 Web 应用程序的域数据进行测试是比较有效的方法，然而，目前已经存在的利用域数据生成测试用例的方法对 Web 使用时的用户交互及 Web 服务器的状态考虑得较少。在今后的工作中将进一步研究 Web 测试方法，以解决多用户交互和服务器状态对功能测试的影响。

参考文献

- [1] Liu Chien-hung, David C K. Structural Testing of Web Applications[C]//Proc. of the 11th IEEE Int'l Symp. on Software Reliability Engineering. San Jose, USA: [s. n.], 2000: 84-96.
- [2] Ricca F, Tonella P. Analysis and Testing of Web Applications[C]//Proc. of Int'l Conf. on Software Engineering. Toronto, Canada: IEEE Computer Press, 2001: 25-34.
- [3] DiLucca G, Fasolino A, Faralli F, et al. Testing Web Applications[C]//Proc. of Int'l Conf. on Software Maintenance. Washington, USA: IEEE Press, 2002: 310-319.
- [4] Sant J, Souter A, Greenwald L. An Exploration of Statistical Models of Automatic Test Case Generation[C]//Proc. of the 3rd Int'l Workshop on Dynamic Analysis. [S. l.]: IEEE Press, 2005.
- [5] Sant J, Souter A, Greenwald L. An Exploration of Statistical Models for Automated Test Case Generation[J]. ACM SIGSOFT Software Engineering Notes, 2005, 30(4): 55-61.
- [6] Elbaum S, Rothermel G, Karre S, et al. Rikant and Marc Fisher II, Leveraging User-session Data to Support Web Application Testing[J]. IEEE Trans. on Software Engineering, 2005, 31(3): 187-201.

编辑 张正兴