

基于 Session 的 Web 应用软件 EFSM 模型构建方法研究

郭俊霞 郭仁飞 许南山 赵瑞莲

(北京化工大学信息科学与技术学院 北京 100029)

摘 要 在 Web 应用软件模型表示研究领域中,研究对象主要为不含 Ajax 技术的应用程序。少数针对 Ajax(Asynchronous JavaScript and XML)的模型构建采用传统的 FSM 模型表示方法,并不能描述客户端消息触发后的参数传递问题;在 FSM 模型的基础上引入 UML 分层模型的表示方法因需要人工干预,不利于测试用例的自动生成。针对上述问题,借鉴 EFSM 模型这一重要的软件描述模型,从用户的 Session 数据出发,通过日志数据分析用户的行为,并记录客户端的操作事件。通过对用户行为和客户端操作事件进行匹配,生成完整的用户会话,从而建立 EFSM 模型。实验结果表明,该 EFSM 模型能够有效地表示 Web 应用程序的状态以及状态的变化情况,并可以有效地为测试用例自动生成服务。

关键词 Web 应用软件,用户 Session 数据,Ajax,EFSM 模型

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.04.034

Study on Construction of EFSM Model for Web Application Based on Session

GUO Jun-xia GUO Ren-fei XU Nan-shan ZHAO Rui-lian

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract In the research field of Web application model, the main research object is the applications which don't include Ajax. A small number of models which consider the specialty of Ajax use the traditional FSM, and can't describe the parameter transfer problems after the message is triggered in client. There is a method in which the UML layered models is introduced into the FSM for Ajax. However, this method needs manual intervention. It is not convenient for the automatic generation of the test case. To address the above problems, this paper presented an approach which builds model for Web applications based on EFSM as an important software description model. This approach firstly analyzes users' behaviors from users-session data which are recorded in server-side. Meanwhile, the client side events will be recorded. Then, the users' behaviors with client-side events are matched to generate the complete user session data, from which the EFSM model can be built for the Web application. The experiments show that the EFSM model built by using the proposed approach can represent the state and its changes of Web applications effectively. The EFSM model is also convenient for generating test cases automatically.

Keywords Web application, User session data, Ajax, EFSM model

1 引言

随着 Web 2.0 时代的到来,很多 Web 应用程序融入了 Ajax 技术。Ajax 是一种用于创建快速动态网页的技术,它使得 Web 应用软件可以动态集成新的组件,即网页的局部刷新会导致网页内部 DOM(Document Object Model)结构发生局部变化。与其他传统的网页相比,异步交互机制使网页的加载速度变快、所传递的数据量变少,并且能够实现局部刷新,因此被广泛应用于现在的 Web 应用中。

对于 Web 应用软件,网页的变化可以表现为 DOM 结构

和内容的改变。对于含有 Ajax 和不含 Ajax 的页面,在运行过程中,其 DOM 结构和内容均可能发生改变;但相比于不含 Ajax 的页面,使用 Ajax 技术的页面由于使用的是异步刷新机制,会导致同一页面中出现多个 HTML 标签可以触发 Ajax 的情况,从而产生不同的 DOM 结构。这也使得在传统 Web 应用软件建模中,将 Web 应用的每一个页面作为模型中的一个状态,状态之间通过超链接、表单提交、构件调用等请求作为迁移的一般方法^[1]不再适用,给基于 Ajax 技术的 Web 应用软件建模提出了巨大的挑战。

扩展有限状态机(Extended Finite State Machine, EFSM)

到稿日期:2017-01-22 返修日期:2017-03-16 本文受国家自然科学基金(61672085)资助。

郭俊霞(1977—),女,博士,讲师,CCF 高级会员,主要研究方向为网络用户行为分析、网络应用程序测试等,E-mail: gjxia@mail.buct.edu.cn (通信作者);郭仁飞(1992—),男,硕士生,主要研究方向为 Web 应用建模和测试,E-mail: guorenfeidz@163.com;许南山(1956—),男,副教授,主要研究方向为过程工业实时监控(系统集成)、基于网络数据库的应用技术研究、计算机网络与 MIS 系统;赵瑞莲(1964—),女,博士,教授,主要研究方向为软件测试、软件可靠性。

模型是一种常用的软件描述模型,它在 FSM 模型的基础上增加了变量、操作以及事件迁移的前提条件,可以更加精确地刻画软件系统的动态行为。因此,针对现在 Web 应用软件的特点,本文从网络用户日志出发,分析用户行为,获取用户在客户端的操作以及客户端与服务器之间的交互,从而建立相应的 EFSM 模型。该模型旨在解决一个网页中多个 HTML 元素触发 Ajax 请求的模型表示问题,以更好地描述客户端网页的状态变化。

本文第 2 节介绍国内外相关研究现状;第 3 节介绍 Web 应用 EFSM 模型的构建方法;第 4 节给出实验设计及具体结果;最后总结全文。

2 相关研究

随着模型语言、模型驱动技术(MDA)和形式化验证技术的逐步成熟,以及以测试为中心的软件开发技术与方法的兴起和应用,基于模型的软件分析技术在学术界和工业界越来越得到重视,同时也为 Web 应用软件测试用例的自动化生成提供了一种非常有效的解决方案^[2],已经成为自动化测试的一个重要研究方向。

文献[3]中以 Web 应用软件的 UML 状态图作为系统的测试模型,并使用 UML 顺序图描述测试目标,通过转换和组合构成 FSM 模型,进而实现测试用例的自动生成以及测试模型和测试用例的可视化等。文献[4]针对 Web 应用软件测试的需求,提出从对象、导航结构、行为 3 个方面分别进行建模的 Web 应用系统组合测试模型。Thummala 等人^[5]给出了一种基于表示层的 Web 应用软件 ASM(Atomic Section Model)模型构建方法,即根据客户端与服务器端的交互,采用动静结合的方法分析 HTML 文件,构建 Web 应用软件的可视化模型。这些方法都是针对传统 Web 应用的,而 Ajax 异步刷新这一特性使得传统的建模方式不再适用。

对于含 Ajax 的 Web 应用建模,Marchetto 等人^[6]给出了一种基于状态的 Web 应用软件建模方法,采用动静结合的分析方法来建立 Web 应用软件的 FSM 模型,模型中的状态为 DOM 实例,状态间的迁移用 Ajax 请求的回调函数表示,由状态模型的语义交互事件产生测试用例。通过动态与静态分析后,该方法得到的 FSM 模型并不十分准确,可能会包含不合理的路径和行为,因此最后需要通过人工检查的方法来增加遗漏的状态和迁移,并删除或修改错误的执行路径。Arora 等人^[7]为了有效地对 Web 应用进行动态检测,结合 FSM 模型和用户 Session 路径来模拟 Web 应用的 DOM 变化,该方法可以准确地获取 Web 应用中由 DOM 操作引起的动态内容的变化。有研究根据 Web 应用程序的客户端爬虫所获取的结果来推断并生成状态流图。文献[8]中通过一个基于 Selenium¹⁾的轻量级 bot 程序分析了 Web 应用,并推断 Web 应用的状态流图。文献[9]通过动态分析 Web 应用程序中用户接口状态的改变,设计爬虫规则,并依据爬虫运行结果生成状态流图。这些方法虽然都对含有 Ajax 技术的 Web 应用进行了研究,但是没有考虑用户事件或服务器消息触发过程中

参数的传递问题。

文献[10]针对 jquery 框架和 CSS 隐藏技术的 Ajax 页面存在的安全性漏洞问题,提出了一种基于模拟用户操作的 Ajax Web 自动化测试模型。该方法通过模拟用户触发 JavaScript 事件,并利用 JavaScript 引擎对该事件进行解析,从而获取相应的动态页面,然后利用 Fuzzing 测试对该事件进行安全性检测,同时分析页面中可能存在的漏洞注入点。

贺涛等人^[11]在 FSM 模型的基础上,对含有 Ajax 部分的网页状态引入了 UML 分层模型。FSM 模型被用来描述不同 URL 网页之间的链接,网页内部的状态迁移由 Ajax 请求触发,采用 UML 状态图进行分层建模,然后将得到的模型手工转换为 Kripke 结构,并进行模型检验。人工干预的环节使得这种分层的 UML 状态图不利于基于模型的测试用例的自动生成。

本文结合基于状态的测试和基于 Session 的测试,从用户 Session 数据出发,解析用户行为,并通过 Web 应用客户端获取应用程序的状态流图;然后通过操作事件与用户行为建立匹配关系,并构建 EFSM 模型。该模型以真实的用户 Session 数据为基础,并结合了客户端的 DOM 状态,能够更好地描述 Web 应用软件在触发 Ajax 操作时的模型表示问题以及状态变化时触发事件的发生条件等问题。

3 Web 应用的 EFSM 模型构建

EFSM 模型中的状态迁移和变量相互关联,迁移之间的变量和操作交互作用,其对应用程序的形式化描述,可以与 Web 应用执行过程中客户端与服务器端的交互相对应。本文根据 EFSM 模型的上述特征,通过用户日志获取用户行为,通过客户端获取触发事件,然后分析客户端与服务器端的交互行为,建立相应的 EFSM 模型。该方法的流程如图 1 所示。本节的后继内容将对流程中的每一步进行详细说明。

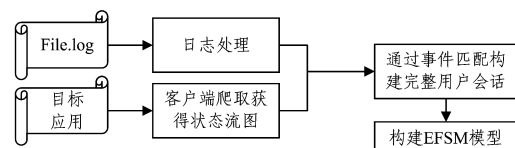


图 1 Web 应用的 EFSM 模型构建方法

Fig. 1 Methods of constructing EFSM model for Web application

3.1 EFSM 模型概述

EFSM 模型是在 FSM 模型基础上进行的扩展。它可以表示为一个六元组:

$$M = (S, s_0, I, O, T, V)$$

其中, S 是一个非空有限的状态集合; s_0 是模型的初始状态; I, O 和 V 分别表示非空的输入事件集合、输出事件集合以及变量集合; T 表示事件的迁移集合,该集合中的一个事件 t 由一个五元组构成,描述为 $t = (Src_t, Tag_t, Event_t, Cond_t, Action_t)$ 。其中, Src_t 和 Tag_t 分别表示事件迁移 t 的源状态和目标状态, $Event_t$ 是 t 的触发事件, $Cond_t$ 表示 t 执行的前提条件, $Action_t$ 表示执行事件迁移 t 后引起的后继操作。

参考 EFSM 模型的定义,本文将 EFSM 模型与 Web 应

¹⁾ <http://www.seleniumhq.org>

用软件进行对应。对于不含 Ajax 的页面和含有 Ajax 的页面,其 DOM 内容在运行过程中均可能产生改变;相比于不含 Ajax 的页面,含 Ajax 的页面由于异步刷新机制会在同一页面产生不同的 DOM 结构。因此,本文重新定义 EFSM 模型中的以下元组。

状态:将 Web 应用软件的网页执行过程中的地址及当前 DOM 结构定义为 EFSM 模型的一个状态。

迁移:将 Web 应用软件网页地址或 DOM 结构发生改变的过程定义为 EFSM 模型的一个迁移,并记录其发生迁移的相关信息,包括触发事件、前提条件和后继操作。

3.2 日志数据的处理

网络用户的日志中记录了大量的用户访问行为,本文将其作为分析用户行为的原始数据。对用户日志的处理主要分为数据清洗、用户识别和会话识别。

由于服务器记录的日志中包含有大量与分析用户行为无关的数据项,如图片、CSS 样式等信息,因此需要对获得的日志进行数据清洗。清洗主要包括对整条记录进行删除和对记录中的无用项进行删除。由于可能存在同一用户在不同时间通过不同的浏览器登录,或者不同用户在同一时间段内使用系统的情况,因此需要对清洗后的日志进行用户识别和会话识别。

用户识别主要是根据日志记录中的访问 IP、用户 SessionID 以及浏览器类型等因素进行分析,对访问该服务器的用户进行识别。会话识别采用的是时间阈值法,即如果一个用户对站点的访问时间超过了阈值 T ,则认为该用户的后续访问记录是一个新会话。本文参考相关研究成果,将阈值 T 设置为 30 min^[12]。通过会话识别,最终可以得到某一用户在不同时间段内对 Web 应用的浏览轨迹,从而分析并提取出用户的事件操作序列。

3.3 客户端行为的获取与分析

在含有 Ajax 的 Web 应用程序中,状态的改变是由用户界面的 DOM 结构来呈现的。客户端调用服务器的事件一旦被触发,就会根据响应来修改页面的 DOM 结构,尤其是与请求紧密联系的响应。因此,通过客户端爬取是获取页面 DOM 结构的一种有效方式。Crawljax¹⁾ 是一个基于 Selenium 的 Ajax 页面爬取工具,它能对大多数页面标签进行处理。本文使用 Crawljax 来获取 Web 页面的 DOM 结构。

由于含有 Ajax 的 Web 应用程序中的多个事件可能会由同一个 JavaScript 函数来处理,因此触发这些事件可能会产生相同的状态记录或 DOM 结构相同但内容不同的状态记录。虽然 Crawljax 在进行 Web 爬取时有约减机制,但由于其采用细粒度约减,因此结果中仍然存在需要进一步处理的页面。例如在一个邮件系统中,收件箱中有 20 封邮件,每页显示 10 封邮件,则第一页和第二页有不同内容的邮件,由于其 DOM 元素中显示的内容不同而会被 Crawljax 忽略,不进行约减处理,但是其 DOM 树结构是相同的则应该进行约减。为了更加有效地检测状态的改变,尽量避免造成状态爆炸,需要对获取到的具有相同 URL 的 DOM 进行比较。

在比较 DOM 之前,首先需要提取 DOM 结构,即过滤与页面结构无关的因素,例如文本内容、时间戳、引用 JavaScript 的语句等。本研究使用 Levenshtein 方法^[13]来进行状态比较,通过计算两棵 DOM 树的编辑距离来判断两个页面的相似程度,并将结构相同的两个状态进行约减,从而避免状态爆炸。由于计算编辑距离时操作对象是页面处理后的 DOM 结构,因此该约减机制适用于大多数的页面。

3.4 用户行为的匹配

一般来说,用户对 Web 应用的操作事件主要包括 Click, MouseOn, MouseOver 等。对于其实现的功能而言,这些事件可以分为单纯的客户端操作事件,如页面被点击后出现弹窗的操作、邮件系统中的选择邮件操作等;或者存在数据交互的操作事件,如登录操作、邮件系统中选中邮件后的删除操作等。这些事件都可以在服务器的日志中进行记录。本文所采用的用户行为提取方法是以获得的用户会话中的事件为主,以从客户端获取的事件为辅,来设计匹配算法,并对从日志中获取的操作序列进行补充,以得到完整的用户操作过程。

事件匹配算法是以 3.2 节中进行数据预处理后产生的用户会话和 3.3 节中生成的状态流图以及网络应用程序本身的 JavaScript 文件作为输入。首先,对状态流图中的某个操作事件,在 js 文件中查找相关操作,如果 js 文件中不存在对该数据的处理或者参数传递的相关操作,即不存在对数据处理的 url 路径和 method, type 等数据处理的数据类型或者参数传递,则认为该操作为单纯的客户端操作事件,记录该事件的源状态、操作事件以及目标状态。然后,分析从目标状态产生的事件,如果该事件在 js 文件中存在数据处理的 url,则将该 url 与通过日志获取的事件访问序列进行匹配,若存在相同的事件,则说明该事件曾经被用户访问过,将该事件与该事件发生前的单纯客户端操作事件均添加到该用户的访问序列中。

算法的伪代码如算法 1 所示。以一个邮件系统为例,假设用户想要删除收件箱中的一封邮件,则其需要操作的事件可以分为两步:操作 1,点击复选框,选中要删除的邮件;操作 2,点击删除按钮进行删除操作。在这个过程中,操作 1 是一个单纯的客户端操作事件,它并没有与后台进行数据交互,因此,该操作在与 js 文件中的信息进行匹配时,不会出现数据处理的 url 等信息;操作 2 进行了数据交互,且该事件会在服务器的日志记录中体现出来。操作 1 和操作 2 的相关信息将分别被添加到用户会话中。

算法 1 用户行为匹配算法

输入:状态流图、JavaScript 文件、用户会话

输出:匹配后的用户会话

方法:

event.init(); // 获取状态流图的第一个操作

while(event != NULL){

if(event isNotIn js) {

在 list 中记录该事件;

} else {

查找处理该事件的 function

¹⁾ <http://crawljax.com>

而实现图书的添加、借阅、归还等操作。表 4 列出了图 2 所示的 EFSM 模型中迁移事件的详细信息。

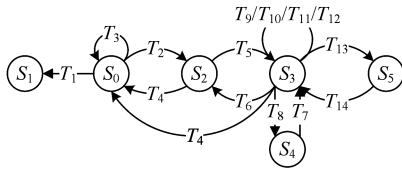


图 2 系统部分功能的 EFSM 模型

Fig. 2 EFSM model of partial function

表 4 EFSM 模型的事件信息

Table 4 Event information of EFSM model

T	Event	Cond	Action	说明
T ₁	click id="window-regist"			点击进入 页面注册功能
T ₂	click id="btnLogin"	Time<=3&& checkValidate(). equals("success")	setAction (true)	登录成功 进入首页
T ₃	click id="btnLogin"	Time>3 checkValidate(). equals("wrong password")		登录失败
T ₄	click id="tool-home-logout"		href="index"	注销
T ₅	click id="book"			进入图书 管理界面
T ₆	click id="return-main"		href="main"	返回首页
T ₇	click id="tool-book-add"			添加图书
T ₈	click id="book-add-confirm"		setAction(true)&& setMessage ("添加成功")	确认添加
T ₉	click id="tool-book-borrow"	row.length>0&& avaliablenum>0	setAction(true)&& setMessage ("借阅成功")	借阅
T ₁₀	click id="tool-book-back"	row.length>0&& num-avaliablenum>0	setAction(true)&& setMessage ("归还成功")	归还
T ₁₁	click id="tool-book-lost"	row.length>0	setAction(true)&& setMessage ("标记成功")	报遗失
T ₁₂	click id="tool-book-delete"	row.length>0	setAction(true)&& setMessage ("删除成功")	删除
T ₁₃	click id="tool-book-edit"	row.length>0		修改
T ₁₄	click id="book-edit-confirm"		setAction(true)&& setMessage ("修改成功")	确认修改

例如,若对图 2 中的路径 $S_0-S_2-S_3-S_4-S_3-S_0$ 生成测试用例,该路径对应的迁移事件分别为 $T_2-T_5-T_7-T_8-T_4$,则 $(S_0, T_2, S_2, T_5, S_3, T_7, S_4, T_8, S_3, T_4, S_0)$ 是一条完整的测试用例。对其进行实例化,然后用自动化测试工具 Selenium 进行验证,结果显示该测试用例是有效的。

为了验证所生成的模型能有效检测 Web 应用中的错误,

本文对源代码人工注入错误后对实验对象进行错误检测(即分析每种类型的错误在源代码中可能出现的位置,并对相应位置的源代码人工注入错误)。文献[8]中指出,Web 应用中的真实错误主要分为 4 类:异步通信错误、交互错误、响应错误、操作转换错误等。本文针对实验对象人工注入不同类型的错误,然后使用基于本文方法构建的 EFSM 模型生成测试用例来对注入的错误进行检测。

在注入的错误中,异步交互错误两个操作之间可能相互影响,如在 Coffee Maker 中,由于时间计数的问题,如果对一个咖啡壶的提示信息不作处理将会影响第二个咖啡壶的相关操作;交互信息是指页面和后台数据交互产生错误,如借阅图书后,图书状态没有发生改变;响应错误指操作过后,不能显示正确的反馈信息;功能性错误是指 Web 应用的某个功能不能正常使用。

检测结果如表 5 所列,其中 Coffee Maker 的响应错误为 2/2,表示共注入响应类型错误 2 个,其中被检测出的错误有 2 个;其余类似。由表 5 可以看出,基于本文方法创建的 EFSM 模型自动生成的测试用例能够检测出大多数错误。未来

表 5 注入错误的检测结果

Table 5 Detection results of injection errors

	Coffee Maker	Board 'R'Us	DBLab
异步通信错误	1/1	1/1	2/2
交互错误	2/2	1/1	1/1
响应错误	2/2	1/1	2/3
功能性错误	1/1	1/1	5/5

结束语 在 Web 应用程序的建模领域中,现有研究主要针对不含有 Ajax 技术的 Web 应用,少数针对含有 Ajax 技术的 Web 应用建模方法,存在不利于测试用例的自动生成或者没有考虑消息触发过程中的参数传递的弊端。本文针对上述问题,从实际 Session 数据出发,通过分析用户日志数据,解析用户行为,获取与行为匹配的事件信息,从而建立 EFSM 模型。实验表明:通过该模型能够有效地表示 Web 应用程序的状态及状态变化情况,并且基于该模型生成的测试用例能够有效地检测 Web 应用程序中存在的错误。未来将主要基于提出的 EFSM 模型进行敏感路径的分析等。

参考文献

- [1] LI L, MIAO H, CHEN S. Test Generation for Web Applications Using Model-Checking[C]// Acis International Conference on Software Engineering, Artificial Intelligence, NETWORKING and Parallel/distributed Computing. IEEE Computer Society, 2010:237-242.
- [2] LEBEAU F, LEGEARD B, PEUREUX F, et al. Model-based vulnerability testing for Web applications[C]// 6th International Conference on Software Testing, Verification and Validation Workshops(ICSTW). IEEE 2013, 2013:18-22.
- [3] MIAO H K, CHEN S B, ZENG H W. Model-based Web application testing [J]. Chinese Journal of Computers, 2011, 34(6): 1012-1028. (in Chinese)
缪淮扣,陈圣波,曾红卫. 基于模型的 Web 应用测试[J]. 计算机学报, 2011, 34(6):1012-1028.

(下转第 214 页)

参考文献

- [1] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 3: 993-1022.
 - [2] MIAO Z, CHEN K, FANG Y, et al. Cost-Effective Online Trending Topic Detection and Popularity Prediction in Microblogging [J]. Acm Transactions on Information Systems, 2016, 35(3): 18.
 - [3] LEE Y J, YEH Y R, WANG Y C F. Anomaly Detection via On-line Oversampling Principal Component Analysis [J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(7): 1460-1470.
 - [4] WU L, HOI S C H, YU N. Semantics-preserving bag-of-words models and applications [J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2010, 19(7): 1908-1920.
 - [5] RAMAGE D, DUMAIS S T, LIEBLING D J. Characterizing microblogs with topic models [C] // Fourth International Conference on Weblogs and Social Media. Menlo Park: AAAI Press, 2010: 130-137.
 - [6] LEE C H, CHIEN T F. Leveraging microblogging big data with a modified density-based clustering approach for event awareness and topic ranking [J]. Journal of Information Science, 2013, 39(4): 523-543.
 - [7] MIKOLOV T. Language Modeling for Speech Recognition [D]. Brno: Brno University of Technology, 2007.
 - [8] MIKOLOV T, KOPECEK J, BURGRT L, et al. Neural network based language models for highly inflective languages [C] // IEEE International Conference on Acoustics, Speech and Signal Processing. Taipei: IEEE, 2009: 4725-4728.
 - [9] TOMAS M, CHEN K, CORRADO G. Efficient estimation of word representations in vector space [EB/OL]. (2013-08-18) [2013-09-07]. <http://arxiv.org/abs/1301.3781>.
 - [10] PEGHOTY. word2vec 中的数学原理 [EB/OL]. <http://blog.csdn.net/itplus/article/details/37969979>.
 - [11] POST 数据集来源网址 [EB/OL]. <http://meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede>.
 - [12] GUO L T, LI Y, MU D J. A Method of Topic Discovery Based on LDA Theme Model [J]. Journal of Northwestern Polytechnical University, 2016, 4(1): 698-702.
 - [13] HUANG B, YANG Y, MAHMOOD A, et al. Microblog Topic Detection Based on LDA Model and Single-Pass Clustering [M] // Rough Sets and Current Trends in Computing. Springer Berlin Heidelberg, 2012: 166-171.
 - [14] GUPTA M, KUMAR P, BHASKER B. Clustering of users on microblogging social media: A rough set based approach [C] // International Conference on Data Science and Engineering. IEEE, 2017: 1-6.
-
- (上接第 207 页)
- [4] LU Y R, WANG F, DENG B. Combined modeling approach for Web application testing [J]. Journal of PLA University of Science and Technology (Natural Science Edition), 2013(6): 617-622.
 - [5] THUMMALA S, OFFUTT J. An Evaluation of the Effectiveness of the Atomic Section Model [M] // Model-Driven Engineering Languages and Systems. 2014: 35-49.
 - [6] MARCHETTO A, TONELLA P, RICCA F. State-Based Testing of Ajax Web Applications [C] // International Conference on Software Testing, Verification, and Validation. IEEE Computer Society, 2008: 3-12.
 - [7] ARORA A, SINHA M. Dynamic content testing of Web Application using user session based state testing [C] // Confluence 2013: the Next Generation Information Technology Summit. 2013: 22-28.
 - [8] BHANU K, PRASANTH S, MOHAN G K. A Bot Driven Framework for Testing Web Applications [J]. Asian Journal of Information Technology, 2016, 15(20): 3905-3911.
 - [9] MANE S S, RAJMANE B A. Automatic Testing of AJAX Applications through Dynamic Analysis of User Interface State Change [J]. International Journal of Computer Applications, 2014, 95(11): 12-16.
 - [10] WANG L N, LI H, ZHAO L. Ajax Web automatic testing model based on simulation of users [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2016, 44(3): 1-5. (in Chinese)
王丽娜, 李怀, 赵磊. 基于模拟用户的 Ajax Web 自动化测试模型 [J]. 华中科技大学学报 (自然科学版), 2016, 44(3): 1-5.
 - [11] HE T, MIAO H K, QIAN Z S. Modeling and Test Case Generation for Ajax-based Web Application [J]. Computer Science, 2014, 41(8): 219-223. (in Chinese)
贺涛, 缪淮扣, 钱忠胜. 基于 Ajax 技术的 Web 应用的建模与测试用例生成 [J]. 计算机科学, 2014, 41(8): 219-223.
 - [12] GUO J X, GAO C, XU N S, et al. User Behavior Analysis Based on Web Browsing Log [J]. Computer Science, 2014, 41(3): 110-115. (in Chinese)
郭俊霞, 高城, 许南山, 等. 基于网页浏览日志的用户行为分析 [J]. 计算机科学, 2014, 41(3): 110-115.
 - [13] MESBAH A, VAN DEURSEN A. Migrating Multi-page Web Applications to Single-page AJAX Interfaces [C] // Euromicro Conference on Software Maintenance & Reengineering. 2007: 181-190.