

基于MVC模型和行为描述的Web应用测试框架

蒋破荒¹, 徐建明²

(1.国家电网湖南省电力公司 信通公司, 湖南 长沙 410007; 2.佳都新太科技股份有限公司, 广东 广州 510665)

摘要: MVC模型分离设计的思想提高了软件的整体质量,然而由于模型只定义了静态架构组件的配置,其并不能明显地提供Web应用程序开发过程中的行为表现,这导致了Web应用程序的描述和其行为特性间较为明显的差异。为了解决这一问题,提出一种行为模型来填补该差异,即对该程序的URL请求序列自动接收机制进行模式化,为Web应用程序提出一个基于行为模式的测试框架。研究表明,当集中于循环行为自动机制时,该框架可提供Web应用程序更好的测试标准。最后,将框架应用于Jakarta Struts框架的配置方案中,其结果表明,该行为模型在对Web应用的测试路径进行分类时取得了较好的效果。

关键词: 计算机软件; MVC构架; Jakarta Struts框架; 软件测试

中图分类号: TN911-34; TP311.5

文献标识码: A

文章编号: 1004-373X(2017)06-0071-04

Web application testing framework based on MVC model and behavior description

JIANG Pohuang¹, XU Jianming²

(1. Communication Company, State Grid Hunan Electric Power Company, Changsha 410007, China; 2. Suntek Technology Co., Ltd., Guangzhou 510665, China)

Abstract: Although the separation design thought of MVC model has improved the overall quality of the software, but because the model only defines the configuration of static architectural components, MVC model can not clearly provide the behavioral expression of Web application program development process, which leads to the obvious differences between the description of Web application program and its behavioral characteristics. To fill this difference, a behavior model is proposed in this paper, in which the automatical receiving mechanism of the URL request sequence of the program is modeled, and a testing framework based on behavior pattern is proposed for Web application program. The research result shows that, when it is focused on the cyclic behavior of an automatic mechanism, this framework can provide a better test standard for Web application program. The result that the framework is applied to the configuration scheme of Jakarta Struts framework shows that the behavior model has achieved a very good effect in the classification of the testing paths of the Web application.

Keywords: computer software; MVC architecture; Jakarta Struts framework; software testing

0 引言

Web应用是计算机响应系统外的网络请求所需的一种软件技术,其基本特征是根据不同交互系统和环境所表现出的特殊行为机制。在传统基于功能模块的软件中,输入输出关系极为重要,这种特征影响着软件架构的各个方面^[1-3]。网页应用普遍建立在基于MVC架构上的J2EE平台上,其为Sun微系统公司制定的一系列构建网页应用的基础API接口集合。MVC模型给出了在API之上来结合组件以开发网页应用的架构模型,其旨在采用合适的方式对组件进行分离式设计,增加Web应用的可维护性和可重用性。然而,由于MVC模型仅

提供抽象的形式,导致了行为描述在描述组件静态架构关系时被分离,在实际开发调试及测试的过程中两者差异较大^[4-6]。因此,本文给出一种简单的行为模型,并定义一种直接行为描述方式。本文的基本想法是将Web应用程序的行为类比为自动机,其状态为JSP,且状态转换被URL请求所调用。假设应用程序有一个进入页面,如欢迎页面,同时有一些最终页面,如感谢页或错误页。在网页应用中,返回到之前访问过的页面常会要求相似的转换。因此,本文考虑将重访页面作为改进测试。基于这种思想,本文给出一种网页程序的测试标准,并通过一个简单的图书管理系统来论证该方法的有效性。

1 网页应用的MVC模型

MVC模型由三种组件构成:模块、视图和控制器。

其中,模块组件代表数据域,视图组件处理客户端的输出,控制器组件定义对数据的操作。MVC模型通常在J2EE平台上使用JSP/Servlet技术^[7],其模型基本结构如图1所示。首先浏览器递交一个需求给控制器,控制器根据需求处理模块组件中的数据,并将处理后的数据更新至后台数据库,然后控制器将会调用一个JSP对请求做出响应。

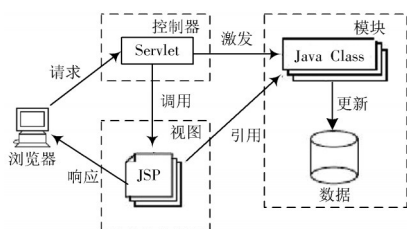


图1 在J2EE平台上使用JSP/Servlet的MVC模型

文中MVC模型使用JSP/Servlet技术,其中模块组件采用JavaBean来实现,视图组件采用Java Server页面,控制器组件则采用ActionServlet实现。J2EE根据指定组件之间连接的配置信息来组装组件。

Struts是由Jakarta项目开发的一个开源网页应用框架,Struts采用遵从如上所述的MVC模型的JSP/Servlet技术,其具体类结构如图2所示。基于MVC模型的行为在其中的实现如下:ActionServlet是一个总的工厂类,负责处理外部请求及分配适当的ActionServlet。开发人员通过定义操作类的子类来实现具体的控制行为。操作类是通过ActionServlet类激发的抽象类。一个Servlet操作指的是一种活动形式目标以及做一些处理操作,其结果被包括到操作转发目标中以返回给Servlet操作^[8-9]。Servlet操作将会根据描述ActionForward到JSP的映射关系ActionMapping来调用一个JSP。最后,所调用的JSP作为响应呈现。与此同时,JSP可能会根据其动态内容做出一些相应处理。

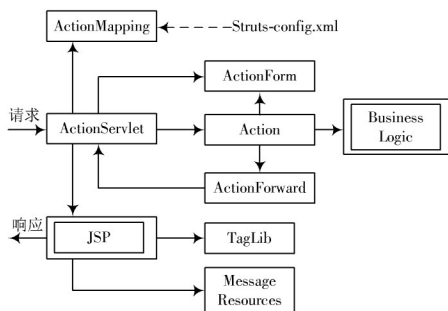


图2 Struts框架

2 行为模型

基于MVC模型的Web应用程序的原始外部行为是

成对的请求序列和JSP,本文将网页应用的行为模式化为一种自动机。鉴于Struts框架,文中用有限状态来完成自动机模式化,其中的状态是一系列出现在配置中的JSP^[10-11]。自动机的字母表来自ActionServlet的请求和响应,这些请求和响应在其中被描述为配置文件。本文假设有一个最初的JSP作为预期行为的切入点,同时假设最终的JSP作为网页应用事务和会话的预期终结点。

定义1 给定一个在Struts中描述的网页应用A,一个有限自动机 $Web_A = (View, Act, AM, iv, FV)$,调用Web自动机定义如下:View是在A中的JSP文件;Act是A的一系列请求及Servlet响应对集合; $AM \subseteq View \times Act \times View$ 是A的操作映射; $iv \in View$ 是A中的初始JSP页面; $FV \subseteq View$ 是A中的最终JSP页面集合。存在 $a \in Act$,本文定义 rq/re ,其中 rq 是一个对ActionServlet的请求标签, re 是对ActionServlet的响应标签,当ActionServlet被激活时, re 被忽略。Action映射指被激活的指向指定的ActionServlet的指令集合,而JSP则根据ActionServlet被调用。

本文的建模目的旨在测试网页应用,即使在最终的JSP中,仍需存在一些链接以开始其另一个请求;但对于测试应用的动态行为,这些链接并不重要,因通常这些链接并不包含应用的逻辑组件,这些请求均惟一地应用映射到含有适当前缀的URL上。为了简单起见,本文忽视这些参数。

图3所示为一个行为模型的示例——简单的图书馆管理系统,其负责管理图书数据库。该图书馆管理系统由三个功能组成,其一是检索系统中已注册的书籍,该功能由JSP,QueryAct构成搜索,由JSP构成列表,并使用未认证的JSP呈现细节。另外两个功能是注册新书以及修改注册记录,这两个功能由JSP执行修改,由ModifyAct及RegistAct执行注册,由JSP显示成功和失败的结果,通过CertifyAct进行认证以及通过JSP来进行登录实现。

图3中Weblib的行为模型组成如下:

$View = \{search, list, detail, login, modify, regist, failure, success\}$

$Act = \{QueryAct/list, QueryAct/detail, CertifyAct/mod, CertifyAct/reg, CertifyAct/NG, ModifyAct/success, ModifyAct/failure, RegistAct/success, RegistAct/failure, link search, link detail\}$

$iv = search$

$FV = \{list, detail, failure, success\}$

$AM = \{(search, QueryAct/list, list), (search, QueryAct/detail, detail), (search, CertifyAct/NG, login), (search, CertifyAct/mod, modify), (search, CertifyAct/reg, regist), (login, CertifyAct/NG, login), (login, CertifyAct/mod, modify), (login, Certi-$

fyAct/reg,regist), (detail, CertifyAct/mod, modify), (detail, CertifyAct/reg,regist), (detail, CertifyAct/NG, login), (modify, ModifyAct/success, success), (modify, ModifyAct/failure, failure), (list, link detail, detail), (regist, RegistAct/success, success), (list, link search, search), (regist, RegistAct/failure, failure), (detail, link search, search), (success, link search, search), (failure, link search, search)}

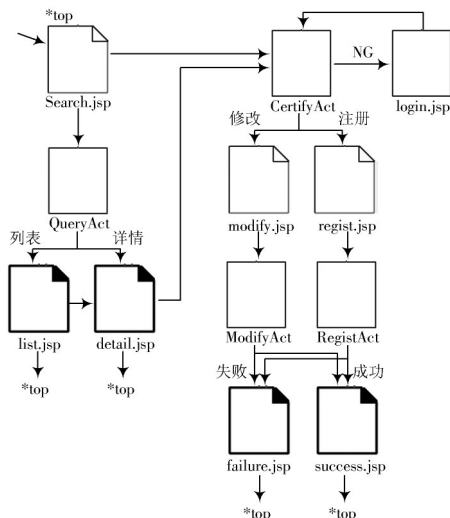


图3 图书馆管理系统

3 测试框架

给定一个网页应用 A , Web_A 将遵从操作映射模拟全部的行为特性。每个字均会被 Web_A 介绍以模仿预期 A 的事务或会话。语言同样也被 Web_A 接收, 即 $L(Web_A)$ 。因此, 检测 $L(Web_A)$ 中的所有字是一种对 A 有效的测试方式。然而, 因 $L(Web_A)$ 是一个常规集合, 通常其是无穷的, 从测试的角度来看, 本文主要关注网页应用的特性, Web_A 中的循环是一个返回之前访问过的页面的序列, 这些重复访问的页面通常会导致应用中的逻辑组件进行相似的应用。据此, 在 $L(Web_A)$ 中最令人感兴趣的测试是其中最多一次循环的字, 为了包含所有转换, 必须将单循环考虑在内。通过增加 JSP 的访问, 可以测试 Web 应用程序更多的属性。本文将这种思想形式化以形成一种测试标准。

Web_{lib} 有两种循环: 由首尾页间链接触发的以及由更新认证链接触发的。前一种循环是相似事务的重复, 后一种循环是用户得到认证前的空循环。给定一个网页自动机 $Web_A = (View, Act, AM, iv, FV)$, 一个执行操作是一个由 iv 引发的 $View$ 和 Act 的交变序列: $v_0 a_1 v_1 \cdots a_i v_i$ 。其中, $v_0 = iv, v_i \in FV$, 且对于所有的 i , $v_i, a_{i+1}, v_{i+1} \in AM$ 。令 $Exec(Web_A)$ 为所有 Web_A 的执行操作计划, $occ(v_i, e)$ 为执行操作 e 中出现 v_i 的次数。文中

记录除去所有来自 e 的 $View$ 活动序列 $L(e)$ 和出现在 e 中的 $View$ 集 $V(e)$

定义2 拥有 n 度的 A 测试标准 T_A^n 定义如下:

$$T_A^n = \{L(e) | e \in Exec(Web_A), \forall v \in V(e): occ(v, e) \leq n + 1\}$$

显然, $T_A^0, T_A^1, T_A^2, \dots$ 是递增数列且 $\bigcup_{n=0}^{\infty} T_A^n = L(Web_A)$ 。

在很多时候, T_A^n 给出了一组大量的测试序列, 图4显示了网页应用的测试框架。

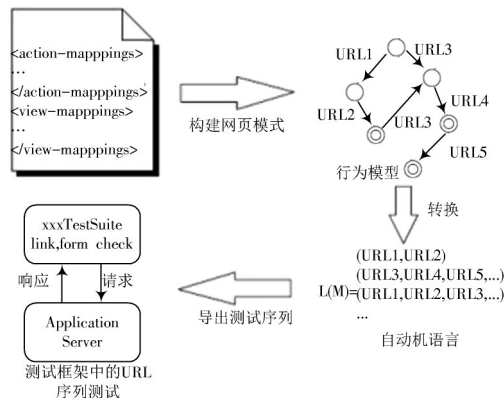


图4 网页应用的测试框架

在这一框架中, 通过一个网页应用配置构造了网页自动机, 然后提取了执行操作。接着根据测试标准进行了测试, 从 T^n 开始直到得到满足条件的结果。测试标准的最大维度取决于测试中的网页应用, 若任何循环均无副作用, 则 T^0 就足够了。

将测试标准应用在 Web_{lib} 中。如前所述, 特征值为 $L(Web_{lib})$ 的正则表达式 Rep_{lib} 如下:

$$Rep_{lib} = (Path_1 + Path_2)(link_search(Path_1 + Path_2))$$

其中,

$$\begin{aligned} path_1 &\underline{\text{def}} CertifyAct/NG * \\ &(CertifyAct/mod(ModifyAct/success + ModifyAct/failure) + \\ &CertifyAct/reg(RegistAct/success + RegistAct/failure)) \\ Path_2 &\underline{\text{def}} (QueryAct/list(e + link_detail) + \\ &QueryAct/detail)(e + Path_1) \end{aligned}$$

此处描述测试标准为 T^0 和 T^1 时的测试序列, 使用 $Rep_{lib} \uparrow T^i$ 来表示语言限制为 T^i , 如下:

$$Rep_{lib} \uparrow T^0 = Path_1^0 + (QueryAct/list(e + link_detail))(e + Path_1^0)$$

其中,

$$\begin{aligned} Path_1^0 &\underline{\text{def}} (e + CertifyAct/NG) * \\ &(CertifyAct/mod(ModifyAct/success + ModifyAct/failure) + \\ &CertifyAct/reg(RegistAct/success + RegistAct/failure) + \\ &QueryAct/detail) \end{aligned}$$

注意到 $CertifyAct$ 的 $ActionServlet$ 被重新激活多次,

但由于其不是JSP,所以不会对测试标准造成影响。

4 Struts的应用

将Struts的配置方案扩展以直接描述行为。主要增加了JSP和动作映射的结合信息。通过这种扩展,可以在Struts中描述一个网页应用的网页自动机。本文为视图组件创建了一个配置文件(wsm-config.xml),该扩展描述了JSP间的转换关系。如图5所示。

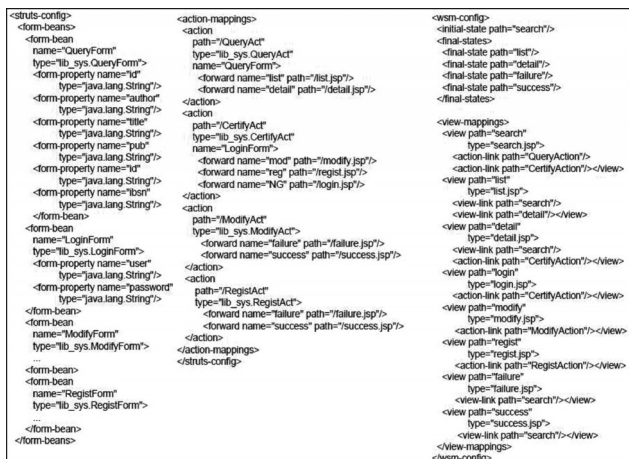


图5 扩展的配置方案

图5显示了图书馆管理系统中的struts-config.xml和wsm-config.xml,网页自动机Web_{lib}由以下部分组成:iv指定为初始状态元素的路径属性,FV则代表了终止状态元素的路径属性。视图和部分AM指定为视图映射元素,每个拥有路径属性的视图元素指定一个视图和由该视图链接的可能状态转换。动作链接元素定义了一个由其路径属性所请求的Servlet调用的状态转换。AM的其余部分由struts-config.xml中的一个动作映射元素所指定。根据每个和活动链接中路径元素所匹配的动作元素,这些元素表明哪一种Servlet活动应被激活以及哪些JSP应被活动Servlet执行的结果所调用。因此,AM是被视图元素中的活动链接元素及映射活动元素所指定的。

5 结论

本文提出了一种基于MVC模式的网页应用测试框架,首先定义了一种网页应用的行为模型,并在J2EE平

台上采用JSP/Servlet技术实现的MVC结构展示了一种面向Jakarta Struts框架的应用。该模型直观地呈现了网页应用全面的行为,且在这种行为模型下进行了测试。本文的重点在于网页应用的特性,即通过重访某些页面经常导致相似的页面转换,为此提出了基于重访问相同页数量的测试标准,相比较其他测试其更为精确。尽管测试标准基于测试中的网页应用的特性,在众多情况下预期最简单的测试标准可涵盖哪些仅重复访问一次的测试序列。

参考文献

- [1] 林艳琴.Web功能测试自动化的研究与应用[D].成都:电子科技大学,2010.
- [2] 侯婷.基于Web功能测试自动化研究与应用[D].青岛:山东科技大学,2014.
- [3] 邓璐娟,范乃梅,孙义坤,等.基于Web应用的性能测试模型研究[J].计算机工程与应用,2013,49(1):75-77.
- [4] 王新.基于MVC模式的通用Web软件系统开发框架设计与实现[D].成都:电子科技大学,2007.
- [5] 郝红岩.基于MVC模式的Web框架的应用研究[D].武汉:武汉理工大学,2013.
- [6] 王俊芳,李隐峰,王池.基于MVC模式的ThinkPHP框架研究[J].电子科技,2014,27(4):151-153.
- [7] ATKINS D L, BALL Thomas, BRUNS Glenn, et al. Mawl: a domain - specific language for form - based services [J]. IEEE transactions on software engineering, 1999, 25(3): 334-346.
- [8] 欧阳宏基,葛萌.基于Struts2与Hibernate框架的Java Web应用研究[J].自动化技术与应用,2016,35(2):48-52.
- [9] MORALES-CHAPARRO R, LINAJE M, PRECIADO J C, et al. MVC web design patterns and rich internet applications [J]. Proceedings of the jornadas de ingenieria del software y bases de datos, 2007, 32(5): 98-104.
- [10] 曾红卫.Web应用的验证与测试方法研究[D].上海:上海大学,2008.
- [11] WOJCIECHOWSKI J, SAKOWICZ B, DURA K, et al. MVC model, struts framework and file upload issues in web applications based on J2EE platform [C]// Proceedings of the 2004 International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science. [S.l.]: IEEE, 2004: 342-345.

作者简介:蒋破荒(1985—),男,重庆人,工程师,硕士。研究方向为电子技术应用。

徐建明(1982—),男,湖北荆门人,硕士。研究方向为软件工程、云计算。