

文章编号:1001-9081(2014)06-1779-04

doi:10.11772/j.issn.1001-9081.2014.06.1779

基于状态迁移的 Web 应用功能测试用例生成方法

张绍康, 王曙燕*, 孙家泽

(西安邮电大学 计算机学院, 西安 710121)

(*通信作者电子邮箱 wsylxj@126.com)

摘要:针对 Web 应用测试方法错误覆盖率低的问题,提出了一种基于状态迁移的 Web 应用功能测试用例生成方法。该方法通过构造页面状态迁移图、事件迁移表和导航迁移表对 Web 应用的链接关系进行表示。基于页面状态迁移图生成的页面状态迁移树生成测试路径,根据等价类划分原则,提出了一种测试用例覆盖准则,结合事件迁移表和导航迁移表中的信息记录构造测试用例。典型案例实验结果表明,该方法有效地表示了 Web 应用中页面间的链接关系,提高了测试用例的错误检测率。

关键词:状态迁移; Web 应用; 覆盖准则; 测试路径; 测试用例

中图分类号: TP311.5 **文献标志码:**A

Test case generation method for Web applications based on state transition

ZHANG Shaokang, WANG Shuyan*, SUN Jiaze

(School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an Shaanxi 710121, China)

Abstract: Due to low error checking rate of Web application test, a method of test case generation for Web applications based on state transition was proposed. By constructing state transition diagram of pages, event transition table and navigation transition table, the link relationship of Web applications was shown. This approach generated test path from state transition tree of pages got from state transition diagram of pages. Based on equivalence partitioning principles, a coverage criteria was proposed, then a test case set was reported as result combined with information from event transition table and navigation transition table. The result shows that the proposed method can represent link relationship of Web applications effectively, and improve error checking rate of test case.

Key words: state transition; Web application; coverage criteria; test path; test case

0 引言

与传统软件相比,Web 应用具有分布式、并发、多用户异构等特点,这些特点对测试提出了新的要求,而测试用例生成是 Web 应用测试的重点^[1-4]。

研究人员针对 Web 应用提出了多种测试方法^[5-12]:文献[7]提出了一种针对各个模块之间交互的 Web 应用测试用例生成方法,该方法通过模块之间的交互关系进行建模,再根据消息传递序列生成相应的测试用例;文献[8]介绍了一种基于 UML 活动图产生测试用例的方法,并考虑了活动中的并发活动的顺序及排序;文献[9]使用错误注入的方法对 Web 应用的安全性进行测试,并给出了 EAS(Environment and Status, 环境与状态错误)模型;文献[10]提出了一种活动图模型驱动的 Web 应用测试方法,从活动图中生成满足覆盖准则的测试路径,基于测试输入语法生成驱动测试序列,再将测试数据整合到测试序列中生成测试用例。但是以上测试方法并没有考虑 Web 应用中页面与页面之间的链接关系,链接是 Web 应用软件的一个主要特征,是 Web 应用功能测试的关键。而文献[11]则是将页面与页面之间的关系统一地看成是一种流的关系,并没有突出它们之间的区别。此外,大多数测试方法

为了生成较少的测试路径而降低了错误检测率。

本文针对以上问题,提出了一种基于状态迁移的 Web 应用功能测试用例生成方法。根据 Web 应用中的页面之间不同的链接关系建立了页面状态迁移图、事件迁移表和导航迁移表,方便了对页面关系的理解和测试用例的生成,降低了 Web 应用的测试成本。此外,根据等效类划分原则提出了一种测试用例覆盖准则,有效地保证了错误检查率。

1 Web 应用的模型

1.1 链接关系

Web 应用分为服务器端和客户端,客户端用于与用户交互,并将用户的请求发送给服务器端,而服务器端则接受请求并产生相应的文档发送给 Web 浏览器。图 1 展示了 Web 应用的工作原理。

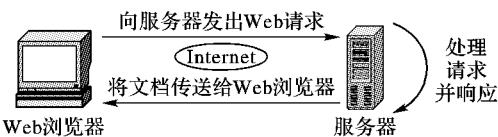


图 1 Web 应用的工作原理

通过 Web 应用的工作原理可得知 Web 应用具有交互性、

收稿日期:2013-11-25; **修回日期:**2014-01-23。 **基金项目:**国家自然科学基金资助项目(61050003);陕西省教育厅自然科学基金资助项目(12JK0732);西安邮电大学青年基金资助项目(ZL2013-26)。

作者简介:张绍康(1989-),男,河南中牟人,硕士研究生,主要研究方向:软件测试; 王曙燕(1964-),女,河南南阳人,教授,博士,主要研究方向:软件测试、数据挖掘; 孙家泽(1980-),男,河南南阳人,副教授,博士研究生,主要研究方向:智能优化算法。

动态性和状态的不确定性。页面之间的跳转反映了页面和页面的链接关系。链接是 Web 应用软件的一个主要特征,一个 Web 应用由多个页面组成,页面和页面之间通过链接进行跳转。Web 应用的页面间的链接关系一般分为两种:导航链接(Navigation Link, NL)和事件链接(Event Link, EL)。导航链接是通过超文本链接实现关联的,而事件链接则是通过触发某个服务器事件实现关联的。

从本质上讲,链接体现了 Web 应用从一个状态到另一个状态的转换,它涉及到 3 个基本要素,即初始状态、状态转换条件和转换后的状态^[9]。正是因为链接的这一特性,可以把 Web 页面视为一种状态,用页面之间的跳转反映状态之间的迁移,将用户的操作视为触发状态迁移的条件。

1.2 相关定义

为了更好地描述本文方法,如下给出了页面状态迁移图、事件迁移表和导航迁移表的定义。

定义 1 页面状态迁移图(State Transition Diagram of Pages, STDP)。是通过将页面视为状态来反映页面和页面之间关系的有向图,它是一个四元组 $STDP = (S, S_0, E_e, E_n)$,其中:

S 是一个有限集合 $\{S_1, S_2, \dots, S_i\}$,称为页面状态集合,集合中元素的数量等于 Web 应用软件中相互关联的页面的数量。

$S_0 \in S$,称为图的初始状态,一般为 Web 软件的起始页面。

E_e 是一个有限集合 $\{e_1, e_2, \dots, e_i\}$,称为事件迁移集合。 e 为图的边, $e_i = \{(S_x, S_y) | S_x \in S \wedge S_y \in S\}$,边所包含的信息记录在事件迁移表中。

E_n 是一个有限集合 $\{n_1, n_2, \dots, n_i\}$,称为导航迁移集合。 n_i 也为图的边, $n_i = \{(S_x, S_y) | S_x \in S \wedge S_y \in S\}$,边所包含的信息记录在导航迁移表中。

定义 2 事件迁移表(Event Transition Table, ETT)。记为 T_e ,它记录了 e 边的迁移信息,包含了边名、起始状态、终点状态、事件名称、所需实际参数列表和跳转页面的 URL。形如表 1 所示。

表 1 事件迁移表

边名	起始状态	终点状态	事件名称	实参列表	URL
e_1	S_4	S_5	register	"zsk", "abcd1234"	./LoginPage.aspx

定义 3 导航迁移表(Navigation Transition Table, NTT),记为 T_n ,它记录了 n 边的迁移信息,它包含了边名、起始状态、终点状态、事件名称和跳转页面的 URL,因为导航迁移并没用事件触发,所以统一为 click。形如表 2 所示。

表 2 导航迁移表

边名	起始状态	终点状态	事件名称	URL
n_4	S_1	S_2	click	./RegisterPage.aspx

从表 2 可以得到导航迁移的全部信息,与表 1 类似。该边的名称为 n_4 ,起始状态和终点状态分别为 S_1 和 S_2 ,事件名称为 click,要跳转的页面 URL 为 $./RegisterPage.aspx$ 。

图 2 给出了一个 Web 软件的 STDP 实例,其中以 e 命名的边表示所连接的页面属于事件链接关系,而以 n 命名的边表示所连接的页面属于导航链接关系。

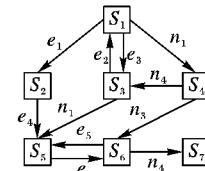


图 2 一个 Web 软件的 STDP 实例

1.3 状态迁移树生成算法

STDP 是一个图形结构,这会给分析带来麻烦,例如,很难判断一条路径是否终止,且该路径可能存在循环,这就使得测试过程变得复杂。因此,本文使用基于 STDP 的页面状态迁移树(State Transition Tree of Pages, STTP)进行测试。STTP 是 STDP 的一个生成树,可以降低问题的复杂性。下面给出 STTP 的生成算法。

算法 1 页面状态迁移树生成算法。

输入 页面状态迁移图 STDP。

输出 页面状态迁移树 STTP。

步骤:

Step1 选取 S_0 为页面状态迁移树的根节点, S_0 一般对应 Web 应用软件中的 HomePage。

Step2 建立一个状态集合 S ,并初始化 $S = \{S_0\}$ 。

Step3 按顺序获取状态集合 S 中的元素 S_i 并标记为访问,即 $S_i.visit = true$ 。若该节点为根节点 S_0 ,则将其能访问的节点 S_k 加入集合 S ,并将 S_k 的父节点标识记为 0,即 $S_k.parent = 0$;若该节点不是根节点,则将该节点能访问的节点中符合判断条件的节点 S_j 加入集合 S 。判断条件为:集合 S 中的节点 S_i 与 S_j 不是同一个节点且父节点标识不同,即 $S_i \neq S_j \& S_i.parent \neq S.j.parent$,并将 S_j 的父节点标识记为 i ,即 $S_j.parent = i$ 。

Step4 重复 Step3 直到集合 S 中所有节点都被访问为止。

Step5 输出页面状态迁移树 STTP。

从图 2 可看出: S_5 和 S_6 , S_1 和 S_3 可能会形成死循环,而算法 1 的 Step3 中的判断条件有效地防止了重复的节点加入到状态集合 S 中,因此避免了死循环的出现。图 3 给出了图 2 的页面状态迁移图生成的页面状态迁移树。

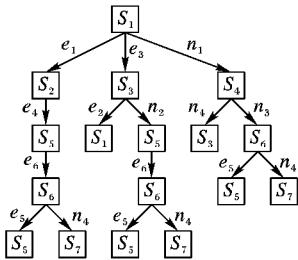


图 3 页面状态迁移树

页面状态迁移树的生成有效地简化了测试过程。首先,从根节点出发,找到其到达每一个叶子节点的路径;然后,记录每条路径上的边的序列,将这些序列汇总起来,就得到了测试路径集 $PS = \{p_1, p_2, \dots, p_i\}$,其中 p_i 表示了一条测试路径。表 3 列出了图 3 的页面状态迁移树获得的测试路径。

表3 测试路径集

名称	测试路径	名称	测试路径
p_1	(e_1, e_4, e_6, e_5)	p_5	(e_3, n_2, e_6, n_4)
p_2	(e_1, e_4, e_6, n_4)	p_6	(n_1, n_4)
p_3	(e_3, e_2)	p_7	(n_1, n_3, e_5)
p_4	(e_3, n_2, e_6, e_5)	p_8	(n_1, n_3, n_7)

2 测试用例生成

2.1 测试用例覆盖准则

为了保证 Web 应用测试的充分性,必须考虑测试用例集合的覆盖准则。比如,“用户密码必须是 8 位且为数字和字母的组合”,那么无法对所有可能的组合进行测试,因此,根据等价类划分原则,本文提出将错误分类的测试用例覆盖准则:

1) 在输入条件规定了取值范围或值的个数的情况下,则可以确立一个有效等价类和两个无效等价类。

2) 在输入条件规定了输入值的集合或者规定了必须如何的条件下,可确立一个有效等价类和一个无效等价类。

3) 在输入条件是一个布尔量的情况下,可确定一个有效等价类和一个无效等价类。

4) 在规定了输入数据的一组值(假定 n 个),并且程序要对每一个输入值分别处理的情况下,可确立 n 个有效等价类和一个无效等价类。

表4给出了遵循测试用例覆盖准则的用户密码的测试用例集。

表4 用户密码的测试用例集

覆盖规则	测试用例	覆盖规则	测试用例
有效的测试用例	“abcd1234”	不符合参数类型	3, 141
不符合参数规定	“abedefg”	不符合参数个数	null

按照测试用例覆盖准则可以生成充分的测试用例集,大多数测试用例生成方法并没有考虑将错误进行分类,而只是笼统地给出了一个有效用例和一个无效用例,此做法不能保证得到较高的错误检测率。

为了保证状态的确定性,在遵循了测试用例覆盖准则后,应增加新的页面状态和新的状态迁移,并对页面状态迁移图、导航迁移表和事件迁移表进行修改和扩充。

2.2 用例生成算法

根据页面状态迁移树,事件迁移表和导航迁移表,就可以生成测试用例了,下面给出测试用例生成算法。

算法2 测试用例生成算法。

输入 测试路径集 $PS = \{p_1, p_2, \dots, p_i\}$ 。

输出 测试用例集 $TS = \{t_1, t_2, \dots, t_i\}$ 。

步骤:

Step1 从测试路径集 PS 中获得一条测试路径 p_i ,并标记该路径为访问过 p_i , $viste = true$ 。

Step2 按顺序取得路径中的每条边 p_i ,若取得边的类型是 e ,即 $p_i.type = "e"$,则转到 Step3,若取得边的类型是 n ,即 $p_i.type = "n"$,则到 Step4。

Step3 按照边的名称 $p_i.name$ 到事件迁移表 T_e 中获得

它的迁移信息,生成子测试用例 ζ_i ,形如 $f(\text{实参类表}, \text{URL})$,其中: f 为事件名称,实参类表是根据参数类表和测试用例覆盖准则生成的; URL 是页面跳转的地址。

Step4 按照边的名称 $p_i.name$ 到导航迁移表 T_n 中获得它的迁移信息,生成自测试用例 ζ_i ,形如 $click(\text{URL})$,其中 $click$ 为导航事件的统一名称,URL 是页面跳转的地址。

Step5 将生成的子测试用例 ζ_i 按顺序组合起来,就得到该条测试路径的测试用例 t_i 。

Step6 重复 Step2 ~ Step5 直到测试路径集 PS 中的所有路径都被访问并生成测试用例为止。

Step7 输出测试用例集 TS 。

3 案例分析

本章利用“软件管理平台”中一个简单的登录注册系统来举例说明本文的测试方法。图4给出了登录注册系统的页面状态迁移图,该系统已经注册了一个用户,名为“admin”,密码为“abcd1234”。该系统的用户密码规则与2.1节的规则相同。

根据2.1节的测试用例覆盖准则可知, e_2 和 e_4 属于错误的情况,所以应对 e_2 和 e_4 的错误进行分类,那么 e_2 的状态条件就会扩充为2种错误情况,即登录的用户名不存在和登录的用户名存在但是密码与用户名不匹配。 e_4 的状态条件就会扩充为表4所示的3种错误情况。图5给出扩充后的登录注册系统页面迁移状态图,图中用 S_i 代替页面名称。表5给出了登录注册系统的事件迁移表,表6给出了登录注册系统的导航迁移表。

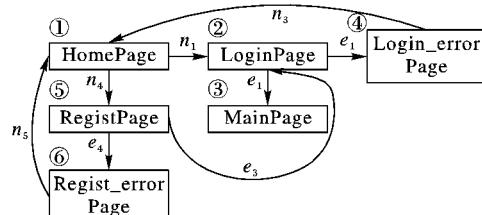


图4 登录注册系统的页面状态迁移图

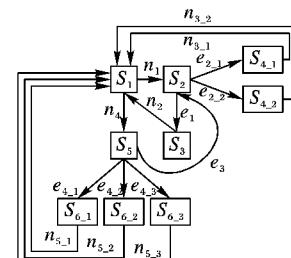


图5 扩充后的登录注册系统页面迁移状态图

有了页面状态迁移图后,就可以生成页面状态迁移树(如图6所示)。

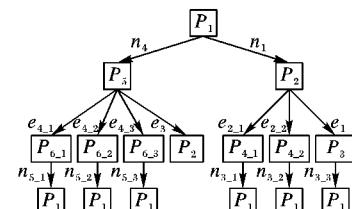


图6 登录注册系统的页面状态迁移树

根据页面状态迁移树,可以得到测试路径并生成测试用例。一共可以得到如下 7 条路径:

(n_4, e_{4_1}, n_{5_1}) , (n_4, e_{4_2}, n_{5_2}) , (n_4, e_{4_3}, n_{5_3}) , (n_4, e_3) ,
 (n_1, e_{2_1}, n_{3_1}) , (n_1, e_{2_2}, n_{3_2}) , (n_1, e_1, n_2)

然后根据事件迁移表和导航迁移表中的迁移信息,就可以得到每条路径对应的测试用例。

路径 1 (n_4, e_{4_1}, n_{5_1}) 对应的测试用例为: click(.. /RegistPage. aspx), regist("zsk", "ABCDEF", .. /Regist_errorPage. aspx), click(.. /HomePage. aspx)。

路径 2 (n_4, e_{4_2}, n_{5_2}) 对应的测试用例为: click(.. /RegistPage. aspx), regist("zsk", 3. 14, .. /Regist_errorPage. aspx), click(.. /HomePage. aspx)。

路径 3 (n_4, e_{4_3}, n_{5_3}) 对应的测试用例为: click(.. /RegistPage. aspx), regist("zsk", null, .. /Regist_errorPage. aspx), click(.. /HomePage. aspx)。

路径 4 (n_4, e_3) 对应的测试用例为: click(.. /

RegistPage. aspx), regist("zsk", "abcd1234", .. /LoginPage. aspx)。

路径 5 (n_1, e_{2_1}, n_{3_1}) 对应的测试用例为: click(.. /LoginPage. aspx), login("zsk", "abcd1234", .. /Login_errorPage. aspx), click(.. /HomePage. aspx)。

路径 6 (n_1, e_{2_2}, n_{3_2}) 对应的测试用例为: click(.. /LoginPage. aspx), login("admin", "11111111"), click(.. /HomePage. aspx)。

路径 7 (n_1, e_1, n_2) 对应的测试用例为: click(.. /LoginPage. aspx), login("admin", "abcd1234"), click(.. /HomePage. aspx)。

用登录注册系统的例子对本文方法和文献[12]方法进行比较,结果见表 7。由表 7 可以知道本文方法产生了比文献[12]方法多的测试用例,而且通过测试用例覆盖准则将错误分类并生成新测试路径,得到了更高的错误检测率。

表 5 登录注册系统的事件迁移表

边名	起始状态	终点状态	事件名称	形参列表	URL
e_1	P_2	P_3	login	"admin", "abcd1234"	.. /MainPage. aspx
e_{2_1}	P_2	P_{4_1}	login	"zsk", "abcd1234"	.. /Login_errorPage. aspx
e_{2_2}	P_2	P_{4_2}	login	"admin", "11111111"	.. /Login_errorPage. aspx
e_3	P_5	P_2	regist	"zsk", "abcd1234"	.. /LoginPage. aspx
e_{4_1}	P_5	P_{6_1}	regist	"zsk", "ABCDEF"	.. /Regist_errorPage. aspx
e_{4_2}	P_5	P_{6_2}	regist	"zsk", 3. 14	.. /Regist_errorPage. aspx
e_{4_3}	P_5	P_{6_3}	regist	"zsk", null	.. /Regist_errorPage. aspx

表 6 登录注册系统的导航迁移表

边名	起始状态	终点状态	事件名称	URL
n_1	P_1	P_2	click	.. /LoginPage. aspx
n_2	P_3	P_1	click	.. /HomePage. aspx
n_{3_1}	P_{4_1}	P_1	click	.. /HomePage. aspx
n_{3_2}	P_{4_2}	P_1	click	.. /HomePage. aspx
n_4	P_1	P_5	click	.. /RegistPage. aspx
n_{5_1}	P_{6_1}	P_1	click	.. /HomePage. aspx
n_{5_2}	P_{6_2}	P_1	click	.. /HomePage. aspx
n_{5_3}	P_{6_3}	P_1	click	.. /HomePage. aspx

表 7 两种测试方法的比较

方法	测试用例数	实际错误种类	生成错误种类	错误检测率/%
本文方法	7	5	5	100
文献[12]方法	4	5	2	40

4 结语

本文根据状态迁移思想建立的页面状态迁移图,事件迁移表和导航迁移表有效地表示并区分了 Web 应用中页面间的链接关系,方便了对 Web 应用结构的理解和测试用例的生成。此外,测试用例覆盖准则将原有的路径进行扩充,从而提高了错误检测率。但是本文方法生成的测试路径存在着冗余的情况。因此,在不降低错误检测率的情况下消除测试路径的冗余是作者下一步的工作。

参考文献:

- [1] STOUT G A. Testing a website: best practices[EB/OL]. [2013-10-10]. http://www.hermotocorp.com/sp-pe/uploads/Annual_Reports/pdf/20130315113443-pdf-68.pdf.
- [2] CNNIC. 中国互联网络发展状况统计报告(2011 年 1 月)[R/OL]. [2012-01-16]. <http://www.cnnic.net.cn/hlfzyj/hlwxbg/201201/P020120709345264469680.pdf>.
- [3] YU L, DU M, ZHANG P, et al. Survey on Web security testing technologies[J]. Application Research of Computers, 2012, 29(11): 4001 - 4005. (于莉莉, 杜蒙杉, 张平, 等. Web 安全性测试技术综述[J]. 计算机应用研究, 2012, 29(11): 4001 - 4005.)
- [4] RICCA F, TONELLA P. Analysis and testing of Web applications [C]// ICSE 2001: Proceedings of the 23rd International Conference on Software Engineering. Piscataway: IEEE Press, 2011: 25 - 34.
- [5] FENG Z, GAO J, ZENG H. Research automation test of Web applications[J]. Computer Engineering and Design, 2010, 31(1): 175 - 178. (冯振华, 高菊, 曾红卫. Web 应用自动化测试的研究[J]. 计算机工程与设计, 2010, 31(1): 175 - 178.)
- [6] TORSEL A M. A testing tool for Web applications using a domain-specific modeling language and the NuSMV model checker[C]// Proceedings of the 2013 IEEE 6th International Conference on Software Testing, Verification and Validation. Piscataway: IEEE Press, 2013: 383 - 390.
- [7] ZHOU X, MIAO H, CHEN S. Modeling and testing Web applications based on functional interaction[J]. Computer Engineering and Design, 2010, 31(14): 3139 - 3142(周晓康, 缪淮扣, 陈圣波. Web 应用功能交互的建模与测试用例生成[J]. 计算机工程与设计, 2010, 31(14): 3139 - 3142.)

(下转第 1787 页)

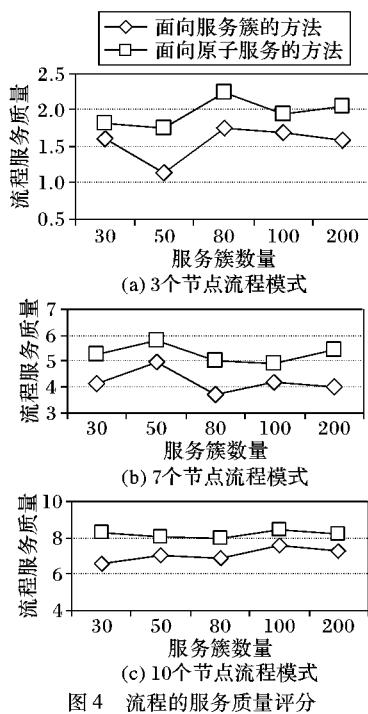


图4 流程的服务质量评分

4 结语

面向服务簇构建服务流程的方法增大了流程中服务响应的粒度,从而缩减了组成服务的查找空间,有效减少了服务流程构建时间。利用逻辑Petri网建模服务流程,给出了流程服务质量的计算方法,在流程推荐服务集的基础上求解质量最佳的服务流程,提高了流程服务质量;在服务推荐集的基础上可以快速进行流程重构,提高了流程响应自适应性。开发了相应的仿真平台,仿真实验证明了该方法的有效性。

参考文献:

- [1] KUZU M, CICEKLI N K. Dynamic planning approach to automated Web service composition [J]. Applied Intelligence, 2012, 36(1): 1–28.
- [2] TAN W, FAN Y, ZHOU M C, et al. Data-driven service composition in enterprise SOA solutions: a Petri net approach [J]. IEEE Transactions on Automation Science and Engineering, 2010, 7(3): 686–694.
- [3] FAN X, JIANG C, WANG J, et al. Random-QoS-aware reliable Web service composition [J]. Chinese Journal of Software, 2009, 20(3): 546–556. (范小芹,蒋昌俊,王俊丽,等.随机QoS感知的可靠Web服务组合[J].软件学报,2009,20(3):546–556.)
- [4] WEN T, SHENG G J, GUO Q, et al. Web service composition based on modified particle swarm optimization [J]. Chinese Journal of Computers, 2013, 36(5): 1031–1046. (温涛,盛国军,郭权,等.基于改进粒子群算法的Web服务组合[J].计算机学报,2013,36(5):1031–1046.)
- [5] da SILVA L M, BRAGA R, CAMPOS F. Composer-Science: a semantic service based framework for workflow composition in e-Science projects [J]. Information Sciences, 2012, 186(1): 186–208.
- [6] WEI L, LI Y, ZHAO Q, et al. Dynamic changing model of workflow process based on adaptive component [J]. Computer Integrated Manufacturing Systems, 2010, 16(12): 2603–2610. (魏乐,李亚玲,赵秋云,等.基于自适应构件的工作流流程动态变更模型[J].计算机集成制造系统,2010,16(12):2603–2610.)
- [7] DORN C, TAYLOR R N, DUSTDAR S. Flexible social workflows: collaborations as human architecture [J]. IEEE Internet Computing, 2012, 16(2): 72–77.
- [8] MAGUIRE T, SNELLING D. Web services service group 1.2 [M]. 2nd ed. London: OASIS, 2006: 1–42.
- [9] LIU X, HUANG G, MEI H. Discovering homogeneous Web service community in the user-centric Web environment [J]. IEEE Transactions on Service Computing, 2009, 2(2): 167–181.
- [10] SHENG Q Z, BENATALLAH B, MAAMAR Z, et al. Configurable composition and adaptive provisioning of Web services [J]. IEEE Transactions on Services Computing, 2009, 2(1): 34–49.
- [11] BENATALLAH B, SHENG Q Z, DUMAS M. The Self-Serv environment for Web services composition [J]. IEEE Internet Computing, 2003, 7(1): 40–48.
- [12] HU Q, DU Y. Service architecture and service discovery oriented to service clusters [J]. Journal of Computer Applications, 2013, 33(8): 2163–2166. (胡强,杜玉越.面向服务簇的服务体系结构及服务发现[J].计算机应用,2013,33(8):2163–2166.)
- [13] DU Y, JIANG C, ZHOU M. Modeling and analysis of real-time cooperative systems using Petri nets [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 2007, 37(5): 643–654.
- [14] DU Y, JIANG C, ZHOU M, et al. Modeling and monitoring of e-commerce workflows [J]. Information Sciences, 2009, 179(7): 995–1006.
- [15] DU Y, JIANG C, ZHOU M. A Petri-net-based correctness analysis of Internet stock trading systems [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 2008, 38(1): 93–99.

(上接第1782页)

- [8] LIU L, WU J. Test case generation for Web applications based on UML activity diagram [J]. Journal of Jiangnan University: Natural Science, 2011, 10(3): 283–288. (刘龙霞,吴军华.基于UML活动图的Web应用测试用例生成[J].江南大学学报:自然科学版,2011,10(3):283–288.)
- [9] DU J, LU Y. Model for vulnerability testing of Web-based application [J]. Journal of Chinese Computer Systems, 2009, 30(12): 2398–2403. (杜经农,卢炎生.一种Web应用软件安全脆弱性测试模型[J].小型微型计算系统,2009,30(12):2398–2403.)
- [10] HE K, LI X, FENG Z. Approach to activity diagram model driven testing for Web applications [J]. Journal of Computer Applications, 2010, 30(9): 2365–2369. (何可,李晓红,冯志勇.活动图模型驱动的Web应用程序测试方法[J].计算机应用,2010,30(9):2365–2369.)
- [11] QIAN Z, MIAO H. Efficient Web software testing method [J]. Computer Science, 2011, 38(2): 152–159. (钱忠胜,缪淮扣. Web软件的一种有效测试方法[J].计算机科学,2011,38(2):152–159.)
- [12] QIAN Z. An approach to generating tests for Web applications based on relation graph [J]. Journal of Chinese Computer Systems, 2013, 34(3): 486–491. (钱忠胜.一种基于关系图的Web应用测试生成方法[J].小型微型计算系统,2013,34(3):486–491.)