基于流程的 Web 应用功能测试模型及其工具的实现①

A Flow-Based Web Application Function Test Model and Implementation of Its Tool

冯雅丽 吴文娟 (华东师范大学 计算机科学与技术系 上海 200241)

摘 要: 本文针对 Web 功能测试缺乏有效的测试模型和自动化工具这一现状，结合流程的概念提出一种 Web 功能测试模型，并基于该模型开发出 Web 功能测试工具。该模型能直观地描述 Web 应用的交互和行为，其支持工具在实践中证明是有效的，能方便、高效地帮助测试人员发现 Web 应用程序中的错误。

关键词: Web 功能测试 测试模型 流程 自动机 测试工具

Web 应用软件测试已经成为当今国内外研究的一个热点，对测试模型的定义、利用测试模型生成测试用例的策略和方法等问题取得初步成果[1-4]。随着 Web 应用程序的广泛应用，对它的测试越来越集中在业务逻辑和表示逻辑方面，Web 功能测试就显得更为重要。本文专注于 Web 功能测试的研究，从面向对象的角度建立起一种描述 Web 功能测试的模型，并基于该模型开发出自动化 Web 功能测试工具 FlowebX。

本文结构安排：第 1 节对相关研究内容进行总结，第 2 节分析 Web 应用功能测试，第 3 节提出基于流程的功能测试模型，第 4 节介绍基于该模型的 Web 功能测试工具的实现，第 5 节通过一个实例展示该套测试方法应用的过程。

1 相关研究

有效的模型是指导测试和提高自动化水平的基础。现有的 Web 测试模型主要有：面向链接与交互等动态内容的结构模型[5]、基于状态图的 Web 应用导航模型[6]、面向对象的 Web 应用测试模型 WTM[1]等。其中，第三种模型是一种较为全面的组合测试模型，它涵盖了 Web 应用的内部结构、动态行为及实体间的关系，但庞大的模型容易引发模型爆炸危机，并且基于模型的后续工作难以开展。

Web 应用功能测试由于涉及众多重复性的动作，因此该阶段主要依靠 GUI 捕获(记录/脚本)和回放工具进行自动化辅助测试，如 MaxQ、WinRunner 以及 IBM 公司的 Rational Robot 等。但这类工具依赖页面设计，由于 Web 应用的 GUI 经常会有变化，被记录的行为需要重新录制才能使用，脚本健壮性大受影响。另外，测试用例往往和所测对象紧密结合在一起，无法实现测试用例的重用。因此，该阶段的测试工具和测试方法自动化水平尚不高，有待于进一步研究。

2 Web应用功能测试的分析

Web 功能测试是根据规格说明来检测 Web 应用是否包含错误的、遗漏的、以及不必要的功能。它区别于结构性测试，同时又与单元测试和集成测试互为补充，具有如下特点：

(1) 将 Web 应用程序看成一个完整的端到端实体进行测试。

(2) 测试人员不必考虑应用程序的内部结构和具体实现。

(3) 在用户界面(如浏览器)上完成。

可以看出 Web 功能测试的任务为逐一地对规格说明中的各功能点进行测试。而一个功能可看作一条完整的、顺序的、根据输入执行预定动作的流程(Flow)。

① 收稿时间:2008-08-18

为测试某一功能，测试人员通过输入各种数据、执行特定动作、观察程序的运行状态以检测其中存在的错误。如果某条流程在经过相对充分(测试本身是难以穷举的)的测试后，每次都能正确地执行到流程的最终接受状态，那么可以判定这条流程代表的功能是按照规范说明的要求实现的。

通过以上分析，Web 应用中的所有功能可以被构建为一系列流程的集合。因此对于 Web 应用功能测试模型的分析归结为对流程的进一步分析。总体上看，流程应具有以下特性：

(1) 目标性：有着明确的目的，即实现规范中的某个功能。

(2) 整体性：由两个或两个以上的有序动作或者步骤组成。

(3) 确定性：测试用例对流程状态的推进是确定的，特定的测试用例对应着确定的预期结果和流程走向。

(4) 层次性：组成流程的活动本身也可能是一个流程，即一条流程里面有可能包含其他“子流程”。

3 基于流程的Web应用功能模型

3.1 Web 应用功能测试模型中的对象划分

本文在 D.C Kung 等人提出的 Web 测试模型[1]基础上，结合功能测试的特征从客户端的角度对 Web 功能测试中所涉及到的实体进行对象划分。考虑 Web 功能测试过程如下：测试人员按照流程步骤逐步提交事务请求，经服务器端处理后将结果返回至客户端，测试人员通过客户端的最终显示来判断 Web 应用程序的功能是否正确实现。

因此，可将 Web 应用程序的实体对象划分为客户端页面和页面中的组件。客户端页面是 HTML 文档，用于在客户端的浏览器上进行显示；组件可以是 HTML 元素、ActiveX 控件、Java Applet 以及与客户端页面或其它组件交互的任意程序模块。

为了表示对象之间的关系，引入以下四种关系类型：聚集(aggregation)、关联(association)、导航(navigation)、迁移(transfer)。其中，聚集和关联关系同于面向对象思想中的聚合和关联关系。导航关系用于表示客户端页面之间的链接关系，而迁移关系用于表示客户端页面提交请求经服务端处理后定向到另一页面的请求响应关系。

利用 Kung 提出的对象关系图(ORD)来描述以上定义的实体对象以及它们间的关系。对于对象关系图和两种引入的关系，给出定义如下：

定义 1. 对象关系图：ORD=(V, R, E)是一个有向图。V：表示对象的节点的集合；R：表示关系类型的标记的集合；$E \subseteq V*V*R$，表示对象之间关系的边的集合。

其中，R={Ag, As, N, T }，Ag 为聚集关系，As 为关联关系，N 为导航关系，T 为迁移关系； $E=\{E_{Ag}, E_{As}, E_N, E_T\}$， E_{Ag} 为对象间的聚集关系，E_{As} 为关联关系，E_N 为导航关系，E_T 则为迁移关系，并且 $E \subseteq V*V*R$。

定义 2. 导航 (navigation/link)：$E_N \subseteq V*V$，是一系列有向边的集合，表示两个客户端页面的导航关系。对于任意两个页面 v1，$v2 \in V$，若有$(v1,v2) \in EN$，则表示客户端页面 v1 可以链接到客户端页面 v2。

定义 3. 迁移(transfer)：$ETV*V$，是一系列有向边的集合，表示客户端页面间的迁移关系。对于任意两个页面 v1，$v2 \in V$，若有$(v1,v2) \in E_T$，则表示客户端页面 v1 向服务器端提交请求，经服务端处理后返回至客户端页面 v2。

3.2 流程自动机(Flow DFA,FDFA)

在介绍本文提出的流程自动机之前，先引入以下定义：

定义 4. 测试用例(Test Case)：为实现某个功能针对每个对象而编制的一组输入数据、执行动作以及预期结果。

定义 5. 测试用例集合(Test Case Set，TCS)：所有对象的测试用例组成的一个有穷非空集合。

定义 6. 测试用例串(Test Case String)：由测试用例集合(TCS)中的元素组成的有穷序列。假定(t1,t2,t3,t4…)为某一流程的用例输入串，其中 t1 为第一个对象的用例输入，t2 为第二个对象的用例输入，以此类推。

定义 7. 测试流程(Test Flow，F)：能够实现软件规范说明中的某个特定功能的对象活动的组合。

流程自动机 FDFA 被定义为一个五元组：FDFA =(V，T，δ，v0，Z)。其中：

V 是对象的有限集合，它的每个元素代表了流程的每一个状态；

T 即为定义 5 中测试用例输入集合 TCS 的简称；

δ：V*T→V 是一个全函数，表示一种从 V*T→V 的映射关系，是为实现软件规范中的某一功能而将流程状态向前推进的对象转移函数，代表了对下一步测试状态的预测。例如，有 δ (v，t)=v’，表示在当前页面 v 下输入测试用例数据 t，经处理后进入下一页面 v’；

v0∈V 是流程的初始页面；

Z∈V 是终态的集合，也即流程终止时所在页面的集合。

定义 8. 功能测试语言 L(F)：流程自动机所接受的测试用例串的集合。

根据该模型，从软件系统构造出一个 FDFA α，并根据规格说明中的功能 FUN，设计出与其相应的测试用例串集合 s。如果测试用例串 s 中的每个元素均能为 α 所接受，则表明 FUN 被正确地实现。

对于一个流程自动机，每个测试用例输入串的设计都是为了实现某个功能，这体现了流程的目标性；每条流程包含若干对象和用例输入，其中任何一个的遗漏或改变都会导致流程的变化，这体现了流程完整性；根据用例输入和转移函数能够唯一地到达下一个状态，体现了流程的确定性；最后，一个接受串的子串有可能也被该流程自动机所接受，这体现了层次性 s。因此可以看出，流程自动机能够体现出上文分析得到的流程概念的四个特性。

3.3 FDFA 的有效性

根据上文可知，FDFA 的构造依据于已实现的 Web 应用程序，而测试用例语言的设计来源于软件规格说明。FDFA 识别的语言 L(F)为正则语言，因此只有当测试用例语言为正则的，FDFA 才是有效的。

功能是由一系列操作步骤以及有限的状态组成：

(1) 设 f 为软件规格说明中描述的任意一个功能，且 f 包含 n(n≥1)个步骤 s1,s2,s3,...,sn，其中 s1 为 f 的开始步骤，f 包含 m(m≥1)个状态 r1,r2,r3,...,rm。

(2) 定义 f 的测试用例集合 C(f)及测试用例串集合 S(f)。

(3) 对于任意的测试用例串 wi ∈ S(f)，wi=t1t2t3...tr(其中 t1,t2,t3,...,tr 属于 C(f))，其用于测试 f 的一系列步骤 si、sj、sk、...、so(1≤i≤n, 1≤j≤n, 1≤k≤n, ..., 1≤o≤n)，并使得 f 到达状态 rl(1≤l≤m)。则有 a(t1,si)=sj, a(t2,sj)=sk, ..., a(tr,so)=rl，其中 a(t,s)=(s'|r)表示当前从步骤 s 输入测试用例 t 到达步骤 s'或者状态 r。

(4) 根据 (1), (2), (3) 可以构造 DFA Nf=({s1,s2,s3,...,sn} ∪ {r1,r2,r3,...,rm}, C(f), a, s1, {r1,r2,r3,...,rm})使得 Nf 识别 S(f)，所以，S(f)是正则的。

(5) 设 f1,f2,f3,...,fn 为软件规格说明中描述的所有功能，则，测试用例语言 L=S(f1) ∪ S(f2) ∪...S(fn)，由(4)的结论以及正则类语言对于并运算的封闭性[7]，可得测试用例语言是正则的，意味着本文构造的 FDFA 模型是有效的。

同时从该模型也可得到以下结论：如果规格说明中所有功能所对应的测试用例串的集合 S 等于流程自动机 α 识别的语言 L(F)，则说明该软件的功能符合规约的描述。L(F)-S 表示已被软件实现、但未被规格说明描述的功能；S-L(F)则表示已在规格说明中描述、但未被软件实现的功能集合。

4 基于FDFA的Web功能测试工具

为方便测试用例的设计，获取能够实际运行的 FDFA 实例，设计出测试用例定义语言和 FDFA 定义语言，并在此基础上实现 Web 功能测试工具 FlowebX。

4.1 测试用例定义语言

测试用例包含着输入数据、执行动作及预期输出。为了实现测试过程的自动化，本文对测试用例涉及到的操作进行抽象，将它们归纳为以下几种：查找对象、设置对象状态(值)、引发对象事件、检测目标对象，结合 DOM 标准提出一种基于顺序结构(Sequence)的测试用例定义语言(Test Case Definition Language, TCDL)，其语法用 EBNF 描述如下：

<S>::={<NAME>;<SET>;<ACTION>;<ASSERT>}

<NAME>::={id string}

<SET>::={[<LD-SET>]|[<LD-SET>;<SET>]}

<LD-SET>::={<LD-STMT>;<SET-STMT>}

<LD-STMT>::={load<ELEMENT-TYPE><FIND-BY> string}

<ELEMENT-TYPE>::={textfield|button|checkbox|form|link...}

<FIND-BY>::={id|name|class|value|alt|index|src|text|...}

<SET-STMT>::=<set string>
<ACTION>::={<LD-STMT>;<FIRE-EVENT-STMT>};
<FIRE-EVENT-STMT>::={fire <EVENT-TYPE>}
<EVENT-TYPE>::={blur|click|doubleclick|focus|submit}
<ASSERT>::={assert <ASSERT-TYPE> string}
<ASSERT-TYPE>::={text|regex}

TCDL 主要包括五种指令(id, load, set, fire, assert)和四种表达式：测试用例命名表达式(NAME)、查找和设置对象状态表达式(SET)、执行动作表达式(ACTION)及检查目标对象状态表达式(ASSERT)。当断言(assert)失败时，说明该测试用例检测出系统的一个失效点。

4.2 FDFA 定义语言

根据上文提到的 FDFA 模型，该语言主要定义 V、T、、v0 以及 Z 五个元素。由于 T 是测试用例集合，可由测试用例定义语言进行定义，所以 FDFA 定义语言主要包括四个部分：状态定义、转换函数定义、开始状态定义以及接受状态集合定义。具体语法描述如下：

<F>::={<DECL-STATE>;<SET-STATE>;<SET-ACCEPT>}
<DECL-STATE>::={state id;[obj id;]<TRANSFERS>}
<TRANSFERS>::=
　　{transfertestcase-id state-id;[<TRANSFERS>]}
<SET-START>::={start state-id}
<SET-ACCEPT>::={accept state-id;[<SET-ACCEPT>]}

该定义语言包含五种指令(id, obj, transfer, start, accept)和四种表达式：DECL-STATE 用于定义 FDFA 中的状态集； TRANSFERS 用于定义转换函数集合；SET-START 用于设置 FDFA 的开始状态，定义中只能出现一次； SET-ACCEPT 则用于定义 FDFA 的可接受状态集合。

4.3 FlowebX 的实现

在定义以上两种语言的基础上，进一步实现流程自动机模型的支持工具 FlowebX。该工具主要由模型分析、测试执行引擎及错误报告三大模块构成。

由于 XML 是一种可把文档按预先定义的语义结构组织起来并进行结构化验证的语言，具备良好的结构和灵活性，而且 TCDL 及 FDFA 定义语言都是简单的正则语言，因此在语言的实现上采用 XML Schema 对 TCDL 和 FDFA 定义语言进行语法约束。模型分析器的功能是从 XML 文件(TCDL 和 FDFA 定义文件)中解析出一个 FDFA 实例。图 1 为 FDFA 模型分析模块的类图。

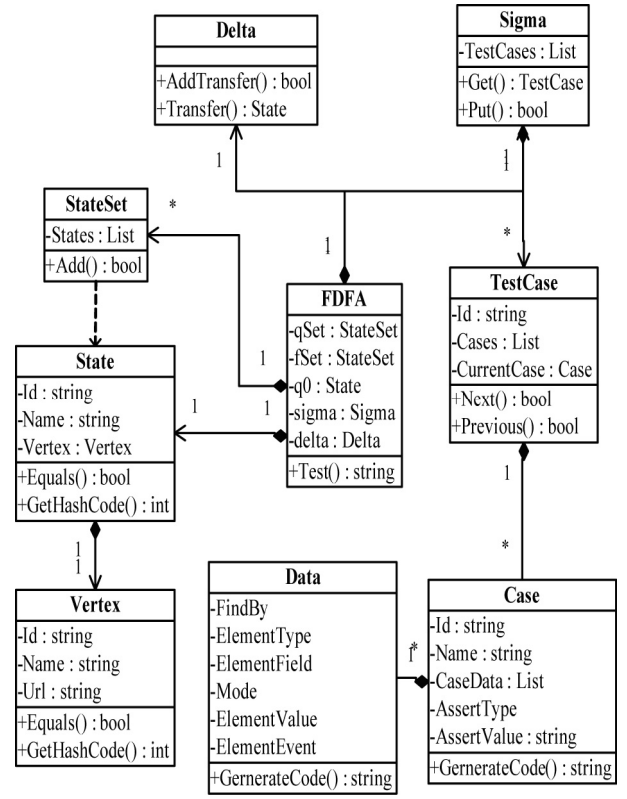

图 1　FDFA 模型分析器的类图设计

测试执行引擎模块根据模型自动生成测试用例串及 FDFA 模型的可执行二进制代码，并对目标 Web 系统执行测试。依据上一模块得到的模型实例，采用一种基于栈的深度优先遍历算法找出模型图中所有可能的测试路径，将测试路径上对应的测试用例进行组合就得到测试用例串。该做法的意义在于减少了人为指定测试用例串易引起的遗漏或错误，增大了测试覆盖率。测试执行引擎根据测试用例串，动态生成 ATrueRunner 类，并调用第三方编译器动态编译该类，最终调用 Run 方法自动执行测试(本文采用开源的 WatiN①框架做为底层的浏览器行为控制模块)。

错误报告模块所报告的错误包括：TCDL 和 FDFA

定义文件中的语法错误、测试执行中所出现的异常错误以及所实现的功能与 FDFA 模型中功能不一致的错误等。定义文件中的语法错误是在编译期间报告的，如果出现此类错误，将无法得到模型实例。对于后两种错误，测试执行引擎将调用 IFlowRunner 接口的 OnError 方法进行报告，错误报告模块通过回调机制获取测试执行引擎所报告的错误并显示给测试人员。

5　实例

由于 Web 应用涉及到的功能一般较为复杂，状态众多，为避免模型过于庞大引发模型爆炸，可以采用分而治之的思想，针对各个模块分别进行模型构建。本节采用的实例是基于一个开源的在线学习平台，涉及到的角色有三类：管理员、教师和学生。所选取的测试功能点为：注册、登录、审核、选课以及创建课程。其中注册和登录为常见场景，面向所有角色开放。审核指的是管理员对教师和学生的注册信息进行审批，选课和创建课程分别为学生和教师的特有功能。图 2 为根据需求规格说明建立起来的相关功能活动图。

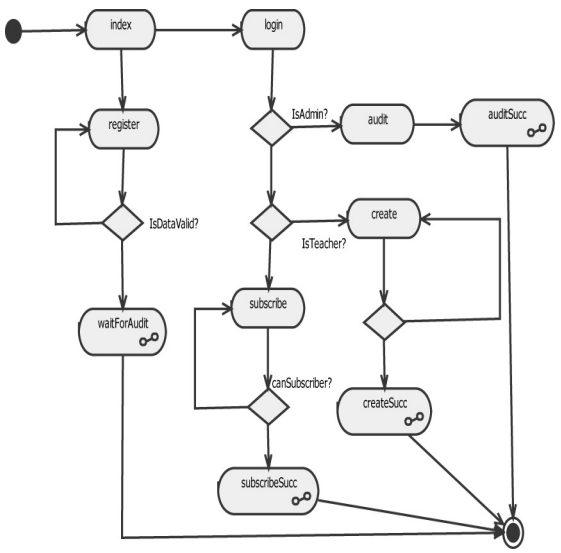

图 2　基于 Web 学习平台的待测功能活动图

FDFA 可构建自活动图，但它的扩展性比活动图好。测试用例则可针对需求说明中对各项的约束以及用例文档中的扩展场景来设计。FlowebX 对测试用例及模型的定义文档进行验证及编译，成功后即可得到待测功能对应的流程自动机模型如图 3，对模型图中的状态及测试用例的说明见表 1。

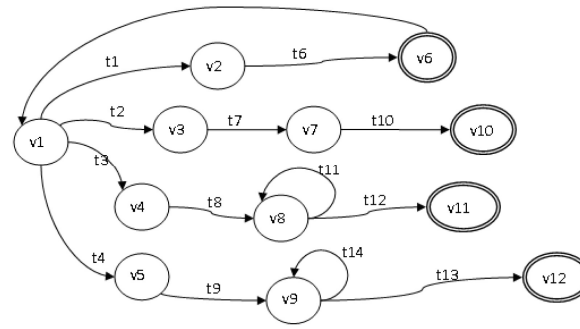

图 3　实例对应的 FFDA 模型图

表 1　FFDA 的状态及测试用例详细对应表

状态	测试用例	目标状态	开始/接受状态
index(v1)	toRegister	v2	开始状态
index(v1)	adminLogin	v3	开始状态
index(v1)	teacherLogin	v4	开始状态
index(v1)	studentLogin	v5	开始状态
register(v2)	register	v6	
adminSpace(v3)	toAudit	v7	
studentSpace(v4)	toCreate	v8	
teacherSpace(v5)	toSubscribe	v9	
registerResult(v6)	toLogin	v1	接受状态
audit(v7)	audit	v10	
createCourse(v8)	invalidCreate	v8	
createCourse(v8)	validCreate	v11	
subscribeCourse(v9)	invalidSubscribe	v9	
subscribeCourse(v9)	validSubscribe	v12	
auditSucc(v10)			接受状态
createSucc(v11)			接受状态
subscibeSucc(v12)			接受状态

FlowebX 生成 4 个测试用例串为[toRegister, register] 、[adminLogin, toAudit, audit] 、[teacherLogin, toCreate, validCreate] 以及[studentLogin, toSubscribe, validSubscribe]。它们分别对应着注册，管理员审核，教师创建课程及学生选课四个功能。根据上文给出的功能描述，可知生成的测试用例串覆盖了全部待测功能。

6 总结

本文提出了一种应用于 Web 功能测试的流程自动机模型，较直观地描述了 Web 应用的行为。模型支持工具实现了测试用例串的自动生成，避免了遗漏不易想到或少用的功能，提高了测试覆盖率。测试用例和模型的分别定义有利于测试用例的重用，因此该工具是一个较为有效的功能测试辅助工具。当然，FDFA 的建模及描述过程还有待于进一步简化，比如，利用开发过程中逐渐形成的 UML 用例图、活动图等自动生成 FDFA 模型。测试用例定义语言也有待于进一步完善和丰富，以便于全面描述各类页面元素及其相应的 DOM 事件。这些都是今后工作中尚需研究和解决的问题。

参考文献

1 Kung DC, Liu CH. An object-oriented web test model for testing web application. Proc. of APAQS, 2000:111－121.

2 Ricca F, Tonella P. Analysis and testing of web application. IEEE Computer, 2001:25－34.

3 Chen SB, Miao HK. Automatic generating test cases for testing web applications. Computational Intelligence and Security Workshops, 2007. International Conference on 15-19 Dec, 2007:881－885.

4 Liu CH, Kung DC, Hsia P. Structural testing of web application. IEEE Computer, 2000:84－96.

5 Ricca F, Tonella P. Analysis and testing of Web application. Proc of the Internation Conference on Software Engineering(ICSE), 2001:25－34.

6 Leung K, Hui L, Yiu S. Modeling Web Navigation by Statechart. Proc of the Computer Software and Applications Conference, 2000:41－47.

7 Michael Sipser.计算理论导引(第二版).北京:机械工业出版社, 2006:19－45.