一种面向对象的 Web Application 测试模型

路晓丽[1,2] 董云卫[2] 赵宏斌[3]
(西北工业大学计算机学院 西安 710072)[1] (西北大学公共管理学院 西安 710069)[2]
(西安供电局信息中心 西安 710032)[3]

摘 要 为了保证Web应用的质量和可靠性,需要不断加强对Web应用软件的测试研究。Web应用软件的有效测试依赖于对其进行充分的分析和理解,提出良好的测试模型,并基于测试模型提出测试策略和测试方法。提出了一种面向对象的Web应用软件测试模型WATM,并且基于WATM提出了测试用例的设计和选择的方法,从而更好地支持Web应用软件的导航测试和状态行为测试。

关键词 Web应用软件,测试模型,Web应用软件测试

Object-oriented Web Application Testing Model

LU Xiao-li[1,2] DONG Yun-wei[2] ZHAO Hong-bin[3]
(College of Computer Science, Northwest Polytechical University, Xi'an 710072, China)[1]
(College of Public Administration, Northwest University, Xi'an 710069, China)[2]
(Information Center, Xi'an Power Supply Bureau, Xi'an 710032, China)[3]

Abstract In order to guarantee Web quality and reliability, people attach more and more importance to Web application testing. Based on good analysis and understanding to Web application, good testing models and methods can be put forward so as to test Web application effectively. An object-oriented Web application testing model and testing methods were offered so as to support navigation testing and state testing.

Keywords Web application, Testing model, Web application testing

1 引言

近几年来,Web应用软件已经成为分布式企业软件系统应用的主流,软件结构越来越复杂,规模越来越大。为了保证Web应用软件的质量和可靠性,需要不断加强对Web应用软件的测试研究。目前,国内外已经提出了一些Web应用软件测试模型,它们之间的侧重点各有不同,各有优缺点。文献[1]提出了一种基于状态图的导航模型来对Web应用进行建模的方法,该方法认为Web应用的页面和页面中的元素可以用状态来表示,页面间的链接和页面中的跳转相当于状态的迁移。不过,此模型试图将Web应用的各种不同的行为都用状态图来描述,使得所得到的状态图变得非常复杂,使用起来并不方便。文献[2,3]提出的面向对象的扩展控制流测试模型可以表达Web应用中实体间的联系,但是该模型只分析了Web应用的控制流信息,而数据流等其他的信息没有涉及,也没有有效地表示各种不同的页面以及页面中的各种组件,也不能表示Web服务器和数据库的交互问题,这对于完整地分析和有效地测试Web应用软件是不够的。文献[4,5]提出的强调链接与交互等动态内容的结构模型集中于Web应用软件的导航特性,所有与结构相关的实体比如链接、表单、框架都被明显地表示出来,但是模型没有考虑页面中包含的脚本、组件和很多内部的实体比如接口对象、服务端页面和被构建页面之间的关系、服务端页面之间的重定向关系等方面,这对于Web应用软件的分析和测试仍然是不够的。文献[6-8]提出了一种面向对象的Web应用测试模型,这种模型较为全面,既考虑了Web应用软件的静态结构的表示,又考虑了动态交互和行为的表示,但是此模型没有考虑多框架页面、页面在不同框架的链接打开和载入及服务端页面重定向页面或者构建客户端页面时对于用户提交数据的依赖,且只能捕获静态数据流,对不同导航场景造成的数据流交互不能检测。

本文在对现有的测试模型进行研究和分析的基础上,结合实践的经验教训,总结整理了相关的工作,提出了一种面向对象测试模型WATM,从而对Web应用软件的静态结构和动态行为进行建模。WATM是基于文献[6-8]提出的,本文对此模型进行了扩充,它考虑了页面在框架中的原始载入和利用target属性在指定的框架中的链接打开,考虑了服务端页面重定向页面或者构建客户端页面时对于用户提交数据的依赖,考虑了不同的导航场景造成的数据流交互的表示,捕获了动态数据流信息。并且基于WATM,进一步讨论了Web应用软件的导航测试和状态行为测试中测试用例的设计和选择的方法。

到稿日期:2009-08-12 返修日期:2009-11-09 本文受国家863计划课题(No. 2009AA01Z147)资助。

路晓丽 博士,副教授,主要研究方向为软件测试;**董云卫** 博士,教授,博士生导师,主要研究方向为嵌入式软件的设计、验证和仿真测试;**赵宏斌** 硕士,高级工程师,主要研究方向为企业服务计算。

2 Web Application 测试模型 WATM

Web 应用软件测试模型(WATM)关注 Web 应用软件与测试相关的特性,并从对象模型和行为模型两方面对软件进行刻画。其中,对象模型将 Web 应用中的实体建模作为对象,并描述它们之间的依赖关系;行为模型包括页面导航图和状态转换图,表示 Web 应用的导航和状态依赖行为。

2.1 对象模型

对象模型将 Web 应用的实体分为 3 类对象,分别为客户端页面、服务端页面和组件,如图 1 所示。客户端页面是可以嵌入脚本的 html 文档;服务端页面是由 Web 服务器执行的 CGI 脚本、ASP、JSP、和 Servlet 等;组件可以是 Java Applet、ActiveX 控件、Html 元素、Java Bean,与客户端页面、服务端页面或者其他组件交互的任意程序模块。为了表示对象之间的关系,除了传统的面向对象体系中的 inheritance, aggregation 和 association 3 种关系外,还需要增加 4 种新的关系类型,分别为 request/submit, response/build, navigation/link 和 redirect,以表示客户端页面和服务端页面间的 HTTP 请求/提交关系、客户端页面和服务端页面间的响应/建立关系、客户端页面之间的导航/链接关系和服务端页面之间的重定向关系。此外,由于每个页面还可以分成框架,因此在表示框架载入初始页面的情况时,将面向对象体系中的关联关系稍作变化,利用给关联关系旁边加"initial page"来表示;而将页面载入不同的框架需要指定链接的目标属性,所以链接关联类 LoadPageFrame 包含了目标框架变量 targetframe 作为可选的数据成员,而 BuildCondition 则指定了依赖输入的动态页面组织和生成的条件。

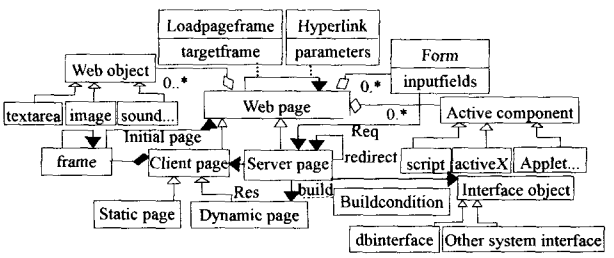

图 1 WATM 的对象元模型

WATM 的对象元模型中可以用对象关系图 ORD 来描述实体以及它们之间的关系[6],该对象关系图 $ORD=(V,R,L,E)$是一个有向图,V 是代表对象的节点集合;R 是对象间关系类型的集合,$R=\{I,Ag,As,Req,Res,N,Rd\}$;$L$ 是标注每种关系的标记的集合;E 是代表对象之间关系的边的集合,$E\subseteq V*V*R*L$,$E=\{E_I,E_{Ag},E_{As},E_{Req},E_{Res},E_N,E_{Rd}\}$,$E_N\subseteq V*V*R*L$,表示两个客户端页面之间的导航关系;$E_{Req}\subseteq V*V*R*L$,表示客户端页面和服务端页面之间的请求关系;$E_{Res}\subseteq V*V*R*L$,表示客户端页面和服务端页面之间的响应关系;$E_{Rd}\subseteq V*V*R*L$,表示两个服务端页面之间的重定向关系。

2.2 行为模型

Web 应用软件的动态行为比较复杂,包括通过超链接从一个页面跳转到另一个页面的导航以及各实体之间状态依赖的行为。可以通过两方面进行建模,一方面利用页面导航图描述 Web 页面之间的导航行为;另一方面利用对象状态图来描述交互对象的状态依赖行为。

(1)页面导航图

页面导航图 $PND=(P,HL,6_0,q_.,F)$是一个有限状态自动机[6],P:所有页面的集合;HL:所有超链接的集合;6_0:转移函数,$6_0=P\times HL-P$;$q_.$:起始页,$q_\in P$;F:终止页集合,$F\subseteq P$。在这个有限状态自动机中,每一个状态表示一个页面,状态之间的变迁表示超链接,并用该超链接的 URL 来标识,每个边可以用 $e_i(i=1,2,\cdots)$来标识。另外,由于一些客户端页面的结构在运行时是动态的,向用户返回何种页面结构取决于 HTTP 请求提交的数据和 Web 应用的内部状态,从而使得同样的超链接会产生不同的客户端页面。为了建模这种情况,放在括号中的条件指定了为激发转移必须为真的被提交数据和内部状态。此外,还必须表示相关的关联类(例如 loadpageframe),并表示相应的动作(例如提交)。

(2)对象状态图

对象状态图 OSD 用来描述 Web 应用中交互对象的状态依赖行为,是一系列层次化的、并发的、交互的状态机。对象状态图 $OSD=(S,p,6,b_0,be)$是一个有限状态自动机[6],s:状态空间;p:Web 页面的集合;b_0:初始状态;be:终结状态;6:变迁函数,$6=S\times p-S$,也可以将其表示为一个三元组(Sa,p,Sb),其含义为:当系统处于状态 Sa 时,运行页面 p 会使系统的状态变为 Sb。每个对象实体的状态是状态空间的向量,各对象不同状态的每一次可能组合便构成了系统的状态。对象之间的交互,例如 HTTP 请求/响应,可以使用"/"后的触发器进行建模,转移的触发改变了对象的状态,对象之间的状态依赖关系用这种触发器形式表示出来;而且为了同步处理并发的对象,在对象状态图中引入了一个 wait 状态,表示对象将会一直等待,直到它其中的一个转移被其他对象触发。

3 基于测试模型 WATM 生成测试用例

测试模型 WATM 中包含了很多 Web 应用的结构和行为信息,基于测试模型 WATM 可以系统地产生行为测试用例,从而支持测试过程的执行。

3.1 导航行为测试

导航行为测试基于 WATM 的行为模型页面导航图设计测试用例,测试用例的产生经过两个阶段来进行,分别为:写出路径表达式和选择测试用例。首先,通过分析页面导航图,构造导航行为图的路径表达式,表达式中的操作符有 3 个,分别为(),+和 *,其中,"()" 用于表示子表达式组,"+"用于路径的选择,"*"用于路径的循环。例如:ele3e5+e2e4e6e7(e8+e9(e10e11)*e12)就是这样的一个路径表达式。接着,从路径表达式中产生测试用例,这时只需要在"+"处拆开路径,并给出循环的具体次数。例如从上例的路径表达式选择的测试用例如下:e1e3e5, e2e4e6e8, e2e4e6e9e10e11e12, e2e4e6e9(e10e11)^{2}e12 等。由于循环的存在,路径很多,这时可以选择比较重要的导航路径进行测试。

3.2 状态行为测试

状态行为测试是基于 WATM 的行为模型对象状态图来设计测试用例,测试用例的产生经过两个阶段进行,分别为:构造测试树和选择路径测试用例。首先,在对象状态图的基础上,将其改造成测试树。测试树中的节点描述了组合对象状态图的复合状态,每个节点的状态用一个 N 元组来表示,

下标 I 代表了是相应对象状态图的第几个状态；树的边代表了状态之间的转移。构造测试树时，树根是所有对象状态图的初始状态的复合状态，根据对象状态图中的节点的转移对树中的节点进行扩展，并且修改相应的状态；直到没有节点需要扩展。其次，遍历测试树来得到测试用例，测试用例是树中起始于根，结束于任何节点的各个路径的转移序列。

4 案例分析

下边给出一个例子来说明如何基于 WATM 来设计测试用例，支持行为测试。

例 1 某个电子商务网站，包含若干个页面，其中，用户可以从页面 login. html 进行登录，login. html 包含一个表单，要求用户输入用户名 username 和密码 password；然后由 verify. asp 进行验证，如果用户名和密码错误，则显示出错页面 error. htmlt，如果用户名和密码正确，则显示主页面 index. html；index. html 由两个框架 $f1$ 和 $f2$ 组成，$f1$ 中的原始页面是商品分类页面 sorts. html，$f2$ 中的原始页面是某类别商品的子类别信息页面 subsort. html；sorts. html 可以在原框架打开一个类别个性显示页面 osorts. html，subsort. html 可以在一个新的窗口（链接 target＝"_blank"）打开商品显示页面 products. html，客户可以在 products. html 中选择喜欢的某一项进行购买；从 products. html 页面可以链接到商品详细信息 productsdetail. html 页面；用户可以在页面 productsdetail. html 中查看商品详细的信息，确定购买后将请求提交到 buy. asp 页面，buy. asp 页面中包含了组件 buyAgent componet 以便处理购买请求。基于处理的结果，会产生相应的页面，即购买活动终止的客户端页面 buyAbort. html 或者购买活动成功的客户端页面 buySucceed. html。从 buyAbort. html 或者 buySucceed. html 页面，用户可以链接到 products. html 页面，继续下一次的购买。而且，buy. asp 页面可以重定向请求给页面 Auth. asp 验证客户身份，并将结果用 Auth. html 发送给用户。

4.1 测试模型

通过对案例的分析，可以得到例 1 的对象模型、导航模型和状态行为模型。

（1）例 1 的对象模型 ORD 如图 2 所示。

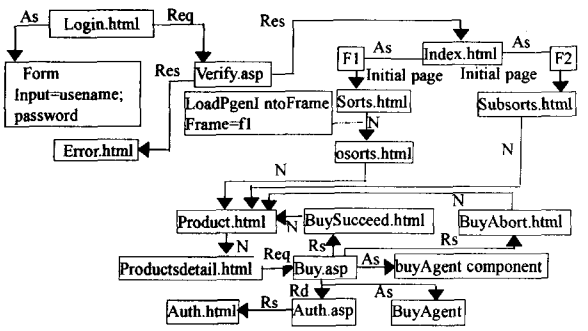

图 2 ORD 例子

（2）例 1 的导航模型如图 3 所示。

（3）对象状态图

例 1 中的客户端页面 productsdetail. html、服务端页面 buy. asp 及所用组件 buyAgent component 相互之间就有状态依赖行为，客户端页面 productsdetail. html 能够接受、重置用户选择的商品信息和输入的身份信息，并提交一个 HTTP 请求给服务端页面 buy. asp；得到请求后，buy. asp 会抽取提交的数据，调用 buyAgent component 进行购买事务处理，并且根据处理的结果返回页面。图 4 是客户端页面 productsdetail. html、服务端页面 buy. asp 及所用组件 buyAgent component 的状态转换图。在图 4 中，submit/S. buy_ request 包含有一个触发器 S. buy_request，它表示如果 submit 转移发生，则 buy. asp 状态 S 中的 buy_request 转移将被触发；wait 状态代表一直等待，直到相应的转移条件被触发。

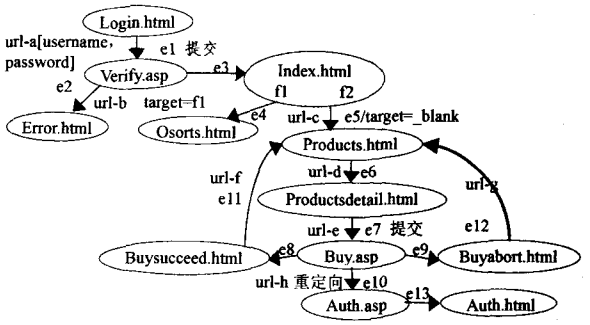

图 3 导航行为图

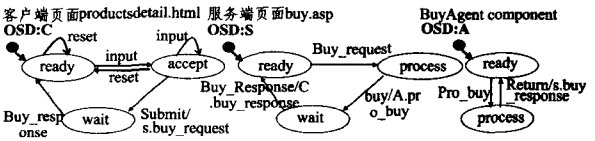

图 4 对象状态图

4.2 基于测试模型设计测试用例

（1）导航测试用例的设计

导航测试用例通过导航行为图得到。遍历例 1 的导航行为图（见图 3），可以得到路径测试用例如下：e1e2；e1e3e4；e1e3e5e6e7e8e11；e1e3e5e6e7e9e12；e1e3e5e6e7e10e13；e1e3e5(e6e7e8e11)^{2}e6e7e9等。由于循环的存在，路径很多，这时可以选择比较重要的导航路径进行测试。

（2）状态行为测试用例的设计

状态行为测试用例利用对象状态图得到。根据例 1 的对象状态图（见图 4），按照状态测试树的构造方法，可以得到测试树如图 5 所示。

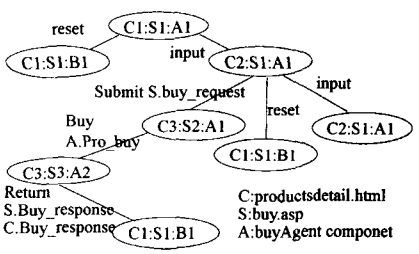

图 5 对象状态测试树

利用图 5，就可以得到 4 个测试用例，如表 1 所列。

表 1 测试用例表

编号	测试用例
1	reset
2	input—(submit, S. buy_request)—(buy, A. pro_buy)—(return, S. buy_response, C. buy_response)
3	Input—reset
4	Input—input

（下转第 151 页）

过程，准确发出漏洞利用的警告。

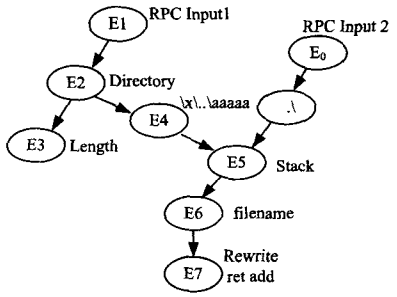

图 6 MS08-067 漏洞污染传播分析

结束语 到目前为止，人们已经提出了许多检测漏洞利用的方法，按照数据截获的位置，这些方法可分成两类：一类是外部阻断型，着重检测从网络接收的各种数据，对数据进行字符、字符串特征过滤，以阻止攻击字符串进入系统；另一类是内部截获型，深入到模块内部甚至内核，监视和检测污染数据，提前发现可能的漏洞利用。基于动态信息跟踪的漏洞利用检测使用污染源标记技术，识别非可信的数据，跟踪并记录污染数据传递的过程，提供污染传播场景信息，对 Open、exec、vfprintf、数据库操作函数、strcmp 等敏感函数参数进行检查，能够检测包括命令注入、格式字符串、缓冲区溢出、SQL 注入、跨站脚本、目录遍历等更加广泛的漏洞利用过程。

参 考 文 献

[1] Kong J, Zou C, Zhou H. Improving Software Security via Runtime Instruction-level Taint Checkingp[C] // Proc. of the 1st Workshop on Architectural and System Support for Improving Software Dependability. Califorria: ACM Press, 2006: 18-24

[2] Lam L, Chiueh T. A General Dynamic Information Flow Tracking Framework for Security Applications[C]// the 22nd Annual Computer Security Applications Conference. Miami Bench, Florida: IEEE Computer Scociety, 2006: 463-472

[3] Newsome J, Song D. Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software[C] // Proc. of the Network and Distributed System Security Symposium. Sandiego California, 2005

[4] Alfond W, Orso A, Manolios P. Using Positive Tainting and Syntax-aware Evaluation to Counter SQL Injection Attacks[C]// Proc. of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York: ACM Press, 2006: 175-185

[5] Pietraszek T, Berghe C. Defending Against Injection Attacks Through Context-Sensitive String Evaluation[C]// Proc. of Recent Advances in Intrusion Detection. Seattle, Washington, 2005

[6] Nguyen T A, Guarnieri S, Greene D, et al. Automatically Hardening Web Applications Using Precise Tainting[C]// Proc. of the 20th IFIP International Information Security Conference. Chiba, Japan, 2005

[7] Suh G, Lee J, Zhang D, et al. Secure Program Execution via Dynamic Information Flow Tracking[C]// Proc. of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM Press, 2004: 85-96

[8] Qin F, Wang C, Li Z, et al. LIFT: A Low-overhead Practical Information Flow Tracking System for Detecting Security Attacks [C]// Proc. Of the 39th Annual IEEE/ACM International Symposium on Microarchitecture. Florida: IEEE Computer Society, 2006: 135-148

[9] Vachharajani N, Bridges M, Chang J, et al. RIFLE: An Architectural Framework for User-Centric Information-Flow Security [C]// Proc. of the 37th Annual IEEE/ACM International Symposium on Microarchitecture. Washington: ACM Press, 2004: 243-254

[10] Leek T, Baker G, Brown R, et al. Coverage Maximization Using Dynamic Taint Tracing[R]. TR-1112. MIT Lincoln Laboratory, 2007

(上接第 136 页)

结束语 为了保证日益复杂的 Web 应用软件的质量和可靠性，需要不断加强对 Web 应用软件的测试研究。实际上，Web 应用软件的有效测试依赖于对其进行充分的分析和理解，提出良好的测试模型，并基于测试模型提出测试策略和测试方法。本文通过对现有测试模型的研究和分析，提出了一种面向对象的 Web 应用软件测试模型 WATM，该模型包括对象模型和导航模型，实现了对 Web 应用软件静态结构和动态行为的有效表示。此外，本文还提出了基于 WATM 来选择和设计导航测试用例和状态测试用例的方法，从而对 Web 应用软件进行导航测试和状态行为测试，更好地保证了 Web 应用软件的质量和可靠性。

参 考 文 献

[1] Leung K, Hui L, Yiu S, et al. Modeling Web Navigation by Statechart[C]// Proc. of Computer Software and Application Conference. 2000

[2] Yang J, Huang J, Wang F, et al. An Objected-Oriented Architecture. Supporting Web Appliction Testing[C]// Proc. of Computer Software and Applications Conference. 1999

[3] Yang Ji-Tzay, Huang Jium-Long, Wang Feng-Jian. A Tool Set to Support Web Application Testing[Z]

[4] Ricca F, Tonella P. Analysis and testing of Web application[J]. IEEE Computer, July 2001

[5] Ricca F, Tonella P. Web site analysis: Structure andevolution[C]// Proceedings of the International Conferenceon Software Maintenance. San Jose, California, USA, 2000: 76-86

[6] Kung D c, Liu C H. An Object-oriented Web test model for testing Web application[J]. IEEE Computer, January 2002

[7] Liu C-H, Kung D C, Hsia P, et al. An object based data flow testing approach for Web applications[J]. International Journal of Software Engineering and Knowledge Engineering, 2001, 11(2): 157-179

[8] Liu Chien-Hung, Kung D c, Hsia P. Structural testing of Web application[J]. IEEE Computer, March 2000