

基于 DataPool 的 Web 测试数据生成与维护方法^{*}

黄 隲^{1,2} 李 诺¹ 金茂忠¹ 刘 超¹

(北京航空航天大学计算机学院 北京 100083)¹ (石家庄陆军指挥学院 石家庄 050084)²

摘 要 针对 Web 应用测试数据所具有的特点,本文提出了一种基于 DataPool 的 Web 应用测试数据生成与维护方法。在形式化定义 DataPool 和明确其语义描述的基础上,根据浏览器端不同的输入域类型在 DataPool 的编辑视图中提供了相对应的测试数据生成方式,并提供了各种维护功能。在 DataPool 浏览视图中,支持以单个和批量的方式选择测试数据。

关键词 DataPool, Web 测试, 测试数据, 数据维护

The Method Test Data Generation and Maintenance of Web Application Based on DataPool

HUANG Long^{1,2} LI Nuo¹ JIN Mao-Zhong¹ LIU Chao¹

(Computer College Beijing University of Aeronautics and Astronautics, Beijing 100083)¹

(Shijiazhuang Army Command College, Shijiazhuang 050084)²

Abstract Aim at the characteristic of Web application test data, this paper propose a method of Web test data generation and maintenance based on DataPool. Based on the formal definition of DataPool and demonstrably describe its semantic, it offer different test data generation method corresponding to input type of Web browser in DataPool editor view, and offer diversified maintenance function. In DataPool browser view, it implement data selector with single and batch mode.

Keywords DataPool, Web testing, Test data, Data maintenance

1 引言

随着 Web 应用越来越普遍,其测试的研究也日益受到关注。目前已有的 Web 测试工具通常集中在验证 html 的语法、静态链接检查、浏览器 GUI 组件的测试以及性能测试等方面^[5],而缺乏针对动态页面的功能测试。Web 应用自动化动态测试的一个核心问题是处理表单的输入数据^[2]。因为动态页面的内容是在运行时依赖于表单的输入数据生成的,因此,为了实施动态测试首先需要解决如何生成、组织与维护这些测试数据的问题。Web 应用的测试数据与传统软件的测试数据相比具有以下特点:

(1)数量大:从横向来看,Web 应用与用户之间的交互完全是依赖于浏览器端的页面,大量的事务处理要通过表单的提交来实现,同时页面之间存在着广泛的链接关系,因此,输入域的数量通常比较多;从纵向看,由于 Web 应用具有分布式和并发性的特点,通常需要一次提供多组测试数据。

(2)类型多:Web 应用的测试数据主要是表单中的输入域数据,为了增强界面的友好性,这些输入域通常采用多种 GUI 组件的形式与用户之间进行交互。

(3)变化快:Web 应用开发中用户的需求经常改变或更新^[7],因此 Web 应用的维护频率比其它软件系统要高,并且在维护过程中通常包含小的增量改变^[6]。因此,Web 应用的回归测试非常重要。同时,为了测试某些新功能,需要随时更新测试数据。

2 相关研究概述

文[1]指出在 Web 测试中应当考虑两种类型,即基于用例的测试和 Web 构件单元测试,并提出了相应的策略,但测试数据的组织问题在该文中没有讨论。

Kohl 等将 Web 页面上的元素均看做对象,然后使用 set 方法由测试人员输入 Text 类型输入域的值^[3],但该方法未区分 Text、Selectbox、Checkbox 以及 Radiobox 等不同的表单输入域类型,同时该方法将测试数据硬编码在代码中也不便于数据的编辑、维护与重用。文[4]提出了形式化的 Web 测试模型,通过 Test Data Composer 来加入测试数据,但该文关注的重点是页面的结构特征,而不关心页面的内容,而在动态页面测试中,我们需要考虑不同的输入所生成的不同的页面。Yang 等使用类似于 BNF 的上下文无关的文法定义测试数据^[5],但该方法只能从形式上定义,其它一些约束信息不易表达,适用于字符串和布尔变量的输入,不适用数值型变量。

文[6]通过解析 server 端的 log 日志文件,得到 user sessions,进而生成测试数据;类似的,Sant 等通过采用机器学习技术,根据 Web log 文件构建马尔可夫模型,从而生成测试数据^[8]。上述方法中得到的测试数据,实际上就是用户的使用数据,而将使用数据存放在 log 文件中的优点是便于数据传输和解析,但不便于对数据进行编辑修改以及实施回归测试。文[9]使用上下文无关的文法来组织测试数据,允许用户在此模式下编辑、删除以及评审测试数据,但该方法并不支持大批

^{*} 基金项目:国家 863 高技术重点项目“网络环境的系统软件核心技术及运行平台”(2004AA119030)。黄 隲 博士后,讲师,主要研究方向为软件工程、指挥自动化。李 诺 博士研究生,主要研究方向为软件工程、软件测试与分析。金茂忠 教授,博士生导师,主要研究方向为软件工程、软件分析与测试、程序语言处理。刘 超 教授,主要研究方向为软件工程。

量的测试数据生成与组织。

从以上分析可以看出,目前有关 Web 测试数据生成的研究大多未考虑重用性,测试数据只使用一次就被丢弃,而对于测试数据的组织维护方法则缺乏相应的研究。为了支持 Web 应用的动态页面测试和回归测试,要求有一种良好的测试数据生成与存储机制,同时要具有较好的可维护性和可重用性。基于此,本文提出了一种基于 DataPool 的 Web 应用测试数据的生成与维护方法,并应用了实际的 Web 应用测试工具中。

3 DataPool 的基本思想与定义

DataPool 是一个值的集合,它可以被表示为一个测试组件或者通过约束被逻辑地描述^[10]。DataPool 是存储与组织大量数据的一种有效方式。Web 应用的使用变化非常快,要求测试数据应具有广泛性;Web 应用的维护率很高,要求测试数据应具有一定的适应性;回归测试在 Web 应用测试中占有非常重要的地位,要求测试数据的可重用性要好。根据上述这些要求,使得在 Web 应用测试中非常适合采用 DataPool 这种机制来生成和组织测试数据。但文^[10]只是提出了 DataPool 的概念,并没有提供实际的解决方案。

3.1 DataPool 的形式化描述

定义 1 DataPool 在表现形式上是一个二维表,其中第一行所有元素构成的集合为 N ,集合中的元素为 $n_i (i = [1..n])$,该集合表示的是一个页面中所有同用户交互的元素的集合;并且有 $N = N_i \cup N_b$,其中 N_i 表示数据输入域的集合, N_b 表示提交按钮的集合;

定义 2 DataPool 中第一列(从第二行开始)所有元素构成的集合为 D ,其中的元素为 $d_i (i = [1..n])$,该集合表示的是和一个页面相对应的所有测试数据的索引的集合;

定义 3 和一个页面相对应的 DataPool 的所有单元格的集合为 C ,其中的元素为 $c_i (i = [1..n])$,则有如下关系: $C = C_i \cup C_c \cup C_r \cup C_s \cup C_b$

定义 4 $C_i \subseteq N \times D$,表示普通文本框和多行文本框类型的输入域;

定义 5 $C_c \subseteq N \times D$,表示复选框类型的输入域;

定义 6 $C_r \subseteq N \times D$,表示单选按钮类型的输入域;

定义 7 $C_s \subseteq N \times D$,表示下拉列表框类型的输入域;

定义 8 $C_b \subseteq N \times D$,表示提交按钮类型的输入域;

定义 9 设含有表单输入的所有页面的集合为 CP ,生成的所有 DataPool 的集合为 DP , $\forall cp_i \in CP (i = [1..n])$,则存在如下映射关系 $F: cp_i \rightarrow dp_i (dp_i \in DP)$ 。

3.2 DataPool 的语义

(1) 一个 DataPool 的规范接口是一个 Collection Interface,实现其接口的数据存储类采用 ArrayList;

(2) 一个 DataPool 中的每一行(即每一组测试数据)作为 ArrayList 中的一个 Element(即一个 Object);

(3) Object 中的 Attribute Value 对应 DataPool 中各列的 Fields Value;若一个 DataPool 中包含 n 组测试数据,则相对应的 ArrayList 的大小为 $n+1$,其中的一个 Constraint Object 用于存储和各列对应的指示信息,该信息表示是否为该列输入了约束条件,该 Object 中的 Attribute Value 的数据类型为 Boolean。

3.3 DataPool 的主要功能

我们知道,Web 应用具有分层的结构,浏览器层、Web 服

务器层、应用服务器层、数据库服务器层等各层之间的信息流通,在单向的范围内是逐层、顺序通信的。在本文中,我们采用了自顶向下的数据产生方式,即通过生成与组织客户端表单中的数据,执行 Web 应用程序,进而生成与输入数据相对应的动态页面以判断其正确性。和 DataPool 相关的主要功能如图 1 所示。

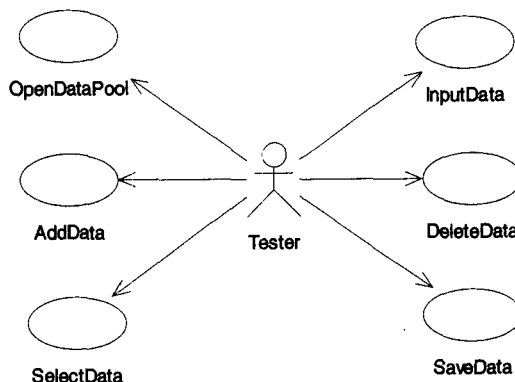


图 1 基于 DataPool 的测试数据生成与维护用例图

在测试过程中,测试人员首先要打开 DataPool 的编辑视图,然后调用 InputData 用例在单元格中输入和编辑测试数据,确认无误后,调用 SaveData 用例将所生成的测试数据保存在元模型中,最后执行 SelectData 用例从生成的所有测试数据中选择需要传输到测试脚本的测试数据。AddData 和 DeleteData 用例负责在 DataPool 中增加和删除测试数据。

4 基于 DataPool 的测试数据生成与维护方法

4.1 不同输入域类型的处理方法

由于客户端页面的所有活动都采用 JavaScript 等技术实现,而 JavaScript 是一种无类型的语言,因此在 DataPool 的处理中可以将所有的输入域都看作 String 类型,其实际的数据类型则放在服务器端来区分。从客户端的角度看,各输入域的不同只是存在于其表现形式上。根据 Web 客户端页面上表单输入域所采取的不同的用户交互方式,将输入域类型划分为文本框类型、复选框类型、单选按钮类型、列表框类型以及提交按钮类型等几种,并在 DataPool 中采取了不同的处理策略,以最大限度地方使用户输入测试数据。

• 文本框类型

对于文本框类型,其提交给服务器的信息完全依赖于用户所输入的数据。因此在 DataPool 中,直接将该类型的输入域设置为一个空白单元格,当单元格处于可编辑状态时,允许用户输入测试数据。

• 复选框类型

此类型即为 GUI 设计中常见的 CheckBox,允许用户在给定的多个选项中选择 0 个或多个选项。为了保持测试数据与输入域的对应关系,在 DataPool 中,每一行只对应着一组特定的测试数据;同时,选项的数量可能会非常庞大。因此,如果在单元格中将所有的选项均显示出来,将会破坏行与列之间的对应关系,并且增加存储开销。

为此,在 DataPool 中将该输入域类型处理为一个 0、1 二进制序列,序列中各位的排序按照表单的 Html 代码的编写顺序确定,其中 0 表示某选项未被选择,1 表示选择该选项。从软件测试的角度考虑,各选项之间是平等的,也就是说我们主要关注的是序列中 0 或 1 的个数,而 0 和 1 的次序则不是

关注的重点。所生成的序列应是若干个具有不同个数的 0、1 序列。根据软件测试的边界值原则,我们在 DataPool 中会为用户自动生成多个有代表性的序列,如全 1 序列(全选),全 0 序列(全不选)等。

• 单选按钮类型

该类型输入域只允许用户在多个选项选择一个作为输入,因此,根据 Html 页面解析的结果提取出所有的选项,将单元格设置为下拉列表框的形式,使得用户可以在其中选择表单中所提供的选项,并且只能选取一个。

• 列表框类型

此类型在表单上的操作形式与单选按钮是类似的,因此,对于该输入域类型的单元格也设置一个下拉列表框供用户输入。

需要指出的是,对于单选按钮和列表框类型的输入,用户既可以在下拉框中进行选择,也可以直接在单元格中输入预定义的值集以外的数据,以测试 Web 应用的容错性和健壮性。

• 提交按钮类型

一个页面中可能包含多个表单,用户点击不同表单中的提交按钮,会在服务器端生成不同的动态页面。因此,为了区分和记录用户的提交操作,在 DataPool 中将页面上的所有提交按钮单独作为一列,并设置成布尔类型的下拉框。其中 true 表示用户点击该按钮,false 反之。

4.2 DataPool 的维护

基于 DataPool 技术,可以在图形交互界面下,采用“所见即所得”的方式实现一系列和测试数据相关的维护操作。

(1)增加数据:为了增强对测试数据的可控性,在初始状态下,DataPool 中当前编辑行的以后各行都是不可编辑的。当用户点击添加按钮后,其后继行变为可编辑状态以追加数据。

(2)删除数据:对于不需要的测试数据,支持单行和多行删除操作。

(3)修改数据:在 DataPool 的编辑视图下,通过单击的方式激活某单元格,直接对测试数据进行修改。

(4)选择数据:文[10]中提出了 Data Selector 的概念,本文将其实体化。DataPool 的浏览视图不同于编辑视图,在浏览模式下用户只能查看所输入的测试数据,而不能对其进行编辑修改,其目的是为了保证所选测试数据和元模型中保存的测试数据的一致性。

在 Eclipse 平台上通过扩展 View 实现的基于 DataPool 的测试数据选择界面如图 2 所示。

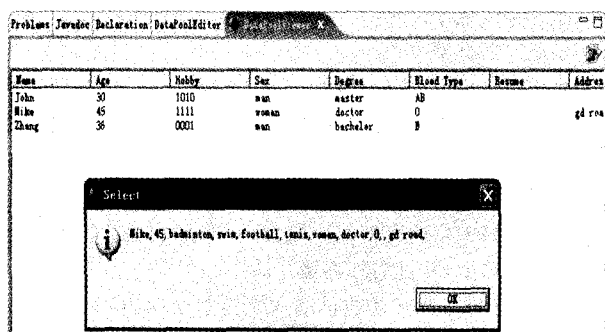


图 2 DataPool 的实现视图

(5)保存数据:DataPool 在保存 CheckBox 类型输入值时,是以 0、1 序列的方式保存的。当在 DataPool 的浏览视图中显示时,要将该序列回映为真实值。

结论及今后的工作 针对 Web 应用测试数据具有的数量大、类型多、变化快等特点,本文提出了一种基于 DataPool 的 Web 应用测试数据生成与维护方法。能够根据浏览器端不同的输入域类型提供相对应的生成方式,最大限度地方便用户输入测试数据,并实现了增加、删除、修改、选择测试数据等功能。今后的工作主要是在用户输入数据的基础上,如何自动化地生成满足要求的测试数据。

参考文献

- 1 Nilawar M. A UML-Based Approach for Testing Web Applications. Master paper of computer science. University of Nevada, Reno. August 2003
- 2 Benedikt M. Juliana Freire Patrice Godefroid. VeriWeb: Automatically Testing Dynamic Web Sites. In: Proceedings of 11th International WWW Conference, Honolulu, May 2002
- 3 Kohl J, Rogers P. Test automation of Web applications can be done more effectively by accessing the plumbing within the user interface. Here is a detailed walk-through of Watir, a tool many are using to check the pipes. Journal of BETTER SOFTWARE. APRIL 2005. 40~45
- 4 Liu Chen-Hung. A Formal Object-Oriented Test Model for Testing Web Applications. Doctor Disseertation. The University of Texas. 2002
- 5 Yang Ji-Tzay, Huang Jiun-Long, Wang Feng-Jian, et al. An Object-Oriented Architecture Supporting Web Application Testing. In: Proc. of IEEE COMPSAC'99, AZ, USA, Oct. 1999. 122~127
- 6 Elbaumt S, Karret S, Rothermel G. Improving Web Application Testing with User Session Data. In: Proceedings of the 25th International Conference on Software Engineering. Portland, Oregon. 2003. 49~59
- 7 Xu Lei, Xu Baowen, Jiang Jixiang. Testing Web Applications Focusing on Their Specialties. ACM SIGSOFT Software Engineering Notes, 2005, 30 (1):1~6
- 8 Sant J, Souter A, Greenwald L. An Exploration of Statistical Models for Automated Test Case Generation. Workshop on Dynamic Analysis (WODA 2005) 17 May 2005, St. Louis, MO, USA
- 9 Yang Ji-Tzay, Huang Jiun-Long, Wang Feng-Jian, et al. Constructing an Object-Oriented Architecture for Web Application Testing. Journal of Information Science And Engineering 18, 2002. 59~84
- 10 UML 2.0 Testing Profile Specification. OMG Adopted Specification ptc/03-08-03