

# 一种基于扩展 WSDL 的测试数据自动生成方法<sup>①</sup>

李盛钢, 丁晓明

西南大学 计算机与信息科学学院, 重庆 400715

**摘要:** 提出一种基于扩展 WSDL 的测试数据自动生成方法用来生成 Web 服务测试数据. 该方法通过扩展 WSDL, 采用等价类划分法、边界值分析法、随机生成法相结合生成简单测试数据以后, 再采用 IPO 方法生成复杂测试数据, 能够有效地防止参数组合爆炸和人为加权造成的测试数据生成不准确.

**关键词:** WSDL; 测试数据; Web 服务; 软件测试

**中图分类号:** TP311

**文献标志码:** A

Web 服务软件测试技术是保证 Web 服务可靠性的关键. 传统的 Web 服务测试方法, 依靠测试人员手工设计测试数据, 这种方法既耗时又具有一定的盲目性. 数目庞大的测试数据决定了自动化测试方法是 Web 服务测试的方向和趋势. 本文提出一种基于扩展 WSDL 的测试数据自动生成新方法, 利用扩展 WSDL 生成简单测试数据, 在简单测试数据之上进一步生成复杂测试数据.

## 1 Web 服务

### 1.1 Web 服务的定义

Web 服务是一种自我描述的、模块化的应用, 一个 Web 服务被部署了, 其他的应用就能够通过网络来发现和调用被部署的 Web 服务<sup>[1]</sup>. 简而言之, Web 服务是一组独立的功能, 这组功能可以通过网络被其他系统轻易调用.

### 1.2 Web 服务的相关技术

Web 服务是通过标准的技术规范生成的. 所有的 Web 服务消息都通过 SOAP(简单对象访问协议)标准的 XML(可扩展标记语言)消息处理协议交换, 然后通过 HTTP 协议实现服务对象的远程调用. 在调用的过程中, 用户根据自己的需求通过 UDDI(统一描述、发现和集成协议)发现符合需求的服务, 最后利用 WSDL(Web 服务描述语言)描述的接口规范编程实现接口<sup>[2]</sup>.

### 1.3 WSDL

WSDL 是一个精确描述 Web 服务的文档, 它将 Web 服务定义为服务访问点或端口的集合. 一个 WSDL 文档通常包含 7 个元素, 即 types, message, operation, portType, binding, port, service 元素<sup>[3]</sup>. 这些元素嵌套在根元素 definitions 元素中. WSDL 主要元素和功能描述如表 1 所示.

① 收稿日期: 2009-12-01

作者简介: 李盛钢(1984-), 男, 湖北随州人, 硕士研究生, 主要从事软件测试研究.

通信作者: 丁晓明, 副教授.

表 1 WSDL 主要元素和功能描述

元素	功能描述
types	数据类型定义的容器, 它一般使用 XML Schema 中的类型系统.
message	使用 types 所定义的类型来定义通信消息的数据结构.
operation	描述服务中所支持的操作, 一般一个 operation 描述一个访问入口的请求/响应消息对.
portType	定义了一种服务访问入口的类型, 它可以包含一个或多个 operation 元素.
binding	定义了特定端口类型、具体网络传输协议和数据格式规范的绑定.
port	定义单个服务访问点的访问入口部署细节.
service	相关服务访问点的集合, 它可以包含一个或多个 port 元素.

## 2 基于扩展 WSDL 的测试数据自动生成方法

基于扩展 WSDL 的测试数据自动生成方法主要分为 3 步: 扩展 WSDL, 简单测试数据生成和复杂测试数据生成. 其结构如图 1 所示.

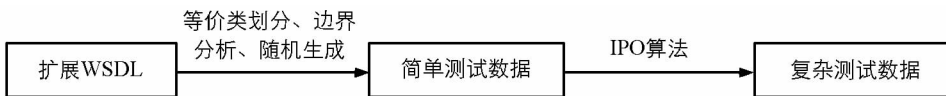


图 1 基于扩展 WSDL 的测试数据自动生成方法

### 2.1 扩展的 WSDL

在 Web 服务中, types 和 message 分别定义了数据类型和消息类型, 它们是测试数据生成的主要依据, 但是, 这两个元素在描述数据类型和消息类型的时候, 不能描述数据的约束关系<sup>[4]</sup>. 因此, 在 types 元素中添加 simpleType 元素<sup>[5]</sup>对数据增加约束关系描述, 这样 WSDL 就能描述数据的约束关系. 扩展以后 WSDL 描述如下所示.

```

<s: element name="number">
  <s: simpleType>
    <s: restriction base="s: int">
      <s: min value="0"/>
      <s: max value="100"/>
    </s: restriction>
  </s: simpleType>
</s: element>
  
```

数据类型主要有数值型、字符型和布尔型三类, 其中, 布尔型只有 true 和 false 两个值, 不存在约束关系; 数值型(int, float, double)和字符型(string)各有 4 个约束关系 max, min, enumeration, pattern. 扩展 WSDL 元素的意义如表 2 所示.

表 2 扩展 WSDL 元素意义

约束关系	数值型	字符型
max	元素最大值	元素最大长度
min	元素最小值	元素最小长度
enumeration	元素值集合	元素值集合
pattern	元素值的正则表达式	元素值的正则表达式

### 2.2 简单测试数据的生成

简单测试数据主要是采用等价类划分、边界值分析和随机生成<sup>[5-8]</sup>相结合的方法来生成. 如果约束关系为 max, min 类型, 首先根据约束关系 max, min 元素将所有待选元素分为大于 max、小于 min 和介于 min 与 max 之间 3 类; 然后根据边界值分析法和随机生成法生成 min-1, min, random(min, max), max,

max+1 测试数据; 最后再增加空值和不符合类型要求的其它类型特殊数据. 各类型相应的简单测试数据生成规则如表 3 所示.

表 3 简单测试数据生成规则

数据类型	简单测试数据
字符型	string(length(min-1)), string(length(min)), string(length(min, max)), string(length(max)), string(length(max+1)), null, random(int)
数值型	min-1, min, random(min, max), max, max+1, null, random(string)
布尔型	true, false

如果约束关系为 enumeration 类型, 生成的简单测试数据为 enumeration 的元素集合; 如果约束关系为 pattern 类型, 生成的简单测试数据为根据 pattern 类型规则随机生成的 5 个元素集合.

生成的简单测试数据分为有效合法值和无效非法值两类, 其中: 测试数据与 WSDL 描述的类型一致且在 WSDL 描述的相关约束关系内属于有效合法值, 其它的均为无效非法值.

### 2.3 复杂测试数据生成

复杂测试数据由一个或者多个简单测试数据组合而成. 它是由简单测试数据根据基于参数顺序渐进扩充的两两组合覆盖测试用例生成方法(IPO)<sup>[9-10]</sup>生成的. IPO 算法的基本思想是以参数为对象, 初始生成满足两个参数组合覆盖要求的复杂测试数据集  $T$ , 然后每次扩展一个参数, 并满足覆盖该参数的所有可能组合, 最终所有参数都扩展到复杂测试数据中去. 在每次扩展参数时, 分为两步: 水平扩展(在原有的复杂测试数据集  $T$  上增加参数进行扩展)和垂直扩展(产生新的复杂测试数据扩展).

简单测试数据是由有效数据和无效数据组成的. 当复杂测试数据生成的时候可能一个复杂测试数据中有多个无效的简单测试数据, 这样的复杂测试数据是没有意义的, 无效的. 一个复杂测试数据最多只能有一个无效的简单测试数据. 为此, 引入了参数约束关系, 它表示两两参数之间的约束关系集合, 即两个参数均为无效简单数据的组合关系集合. 改进后 IPO 的算法如下所示:

输入: 参数  $p_1, p_2, \dots, p_n$  及相应的值集  $T_1, T_2, \dots, T_n$

输出: 复杂测试数据集  $T$

算法:

Start:

$T = \Phi; C = \{\text{两两参数约束关系集合}\}; //$  初始化

$T = \{(v_1, v_2) \mid v_1 \in T_1, v_2 \in T_2, (v_1, v_2) \notin C\}; //$  对于前两个参数  $p_1, p_2$  进行参数覆盖

for  $i = 3$  to  $n$  // 扩展参数, 一次扩展一个参数

{

$A = \{\text{参数 } p_i \text{ 与其它参数 } p_1, p_2, \dots, p_{i-1} \text{ 的两参数组合集, 其中 } p_i \text{ 为必选参数}\};$

$A = A - C;$

for  $j = 1$  to  $|T|$  // 水平参数扩展; 其中  $|T|$  为复杂测试数据集个数

{

从  $T_i$  中取个最佳值来扩展  $T$ , 使得扩展后  $T_i$  的覆盖集  $F$  覆盖  $A$  中组合数最多且  $\forall x$  满足  $x \in F, x \notin C; //x$  为扩展参数后覆盖参数组合

$A = A - F;$

}

if  $A \neq \Phi$  // 垂直参数扩展

{

Add  $T = \Phi; //$  新增测试集

```

For each item in A //item 为 A 中的组合, item 包含了参数  $p_k (1 \leq k < i)$  的值  $v_m$  和参数  $p_i$  的值  $v_i$ 
{
  if AddT 存在一个简单测试数据满足  $p_i = v_i, p_k = \text{"_"}$ , 并且  $v_m$  不会与其它参数形成约束关系
  {
     $p_k = v_m$ ;
  }
  else
  {
    产生新测试数据 NewT, NewT 中  $p_k = v_m, p_i = v_i, p_w = \text{"_"}$  (其中  $w | 1 \leq w < i, w \neq k, w \neq i$ );
    AddT = AddT  $\cup$  NewT;
  }
}
}
T = T  $\cup$  AddT;
}
将 T 中的 "_" 替换成有效的简单测试数据;
输出 T;
End

```

### 3 实 验

本文在理论研究的基础上, 运用 Visual C. NET 开发了一个基于扩展 WSDL 的测试数据自动生成工具 WSTDGT(web service test data generation tool, web 服务测试数据生成工具). 本实验首先编写出多个简单的 Web 服务, 然后对自动生成的 WSDL 进行扩展, 进而生成简单测试数据和复杂测试数据, 最后将自动生成的测试数据作用于预先植入错误的 Web 服务中进行验证. 实验的结果如表 4 所示, 其中: 参数个数表示 Web 服务中参数的个数; WSTDGT 数据表示通过扩展 WSDL 的方法生成简单测试数据后通过 IPO 算法生成复杂测试数据的个数; 组合数据个数表示生成简单测试数据后通过笛卡尔积生成复杂测试数据的个数; 错误总数表示预先植入 Web 服务中的错误总数; 发现错误数表示把利用 IPO 算法生成的复杂测试数据作用于预先植入错误的 Web 服务后发现的错误个数.

表 4 实验结果

参数	WSTDGT 数据	组合数据	错误总数	发现错误数
2	21	36	2	2
3	30	72	2	2
5	47	432	2	2
6	56	2592	4	3
8	74	15542	4	4
9	79	93312	4	4
10	86	559872	5	4

通过表 4 可以看出本文提出的方法可以有效地防止参数组合爆炸, 约简测试数据个数, 参数越多优势越明显, 同时, 生成的测试数据也能够有效检测到 Web 服务中的错误. 在实验过程中, 利用 IPO 算法生成复杂测试数据时, 有效防止了人为加权和加权约简带来的不准确性和不稳定性.

**参考文献:**

- [1] KREGER H. Web Services Conceptual Architecture [R]. New York: IBM Software Group, 2001.
- [2] 岳 昆, 王晓玲, 周傲英. Web 服务核心支撑技术研究综述 [J]. 软件学报, 2004, 3(15): 429 - 442.
- [3] ERIK C, FRANCISCO C, GREG M, et al. Web Services Description Language (WSDL) 1.1[EB/OL]. (2001 - 3 - 15)[2009 - 11 - 15]. <http://www.w3.org/TR/wsdl>, W3C Note.
- [4] 袁雪莉, 丁晓明. 基于扩展 WSDL 的测试用例自动生成 [J]. 西南大学学报: 自然科学版, 2009, 34(2): 140 - 144.
- [5] HANNA S, MUNRO M. An Approach for Specification-based Test Case Generation for Web Services [C]//2007 IEEE/ACS International Conference on Computer Systems and Applications. Washington: IEEE, 2007: 16 - 23.
- [6] SNEED H M, HUANG S. WSDLTest-A Tool for Testing Web Services [C]//Eighth IEEE International Symposium on Web Site Evolution. Washington: IEEE, 2006: 14 - 21.
- [7] 姜 瑛, 辛国茂, 单锦辉, 等. 一种 Web 服务的测试数据自动生成方法 [J]. 计算机学报, 2005, 28(4): 568 - 577.
- [8] 聂长海. 基于接口参数的黑箱测试用例自动生成算法 [J]. 计算机学报, 2004, 27(3): 322 - 388.
- [9] TAI K C, LEI Y. A Test Generation Strategy for Pairwise Testing [J]. IEEE Transactions on Software Engineering, 2002, 28(1): 109 - 111.
- [10] 曾劲涛, 陈建明. 有参数约束的两两组合覆盖测试用例生成的研究 [J]. 苏州大学学报: 自然科学版, 2008, 24(1): 45 - 49.

## A Method of Automatic Test Data Generation Based on Extended WSDL

Li Sheng-gang, DING Xiao-ming

*College of Computer and Information Science, Southwest University, Chongqing 400715, China*

**Abstract:** Based on extended WSDL, the paper presents a method of automatic test data generation for Web Servers. After extending WSDL, this method combine equivalence partitioning, boundary value analysis and randomly to generate simple test data, and by using the method of IPO to generate complex test data. This method can effectively prevent the combinatorial explosion and solve the inaccuracy caused by artificially weighted.

**Key words:** WSDL; test data; wservices; software testing

责任编辑 张 桢