

Web 应用软件的结构测试研究

路晓丽^{1,2} 董云卫¹

(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

摘 要 为了保证 Web 应用的质量和可靠性,Web 应用软件测试受到了人们越来越多的重视。有效的 Web 应用软件的测试依赖于对其进行充分的分析和理解,掌握其内部的控制流和数据流信息,构建结构模型实施基于覆盖率的结构性测试。提出了一种 Web 应用软件结构模型,它分为 5 级,分别为函数级、函数群级、对象级、对象群级和应用级,且基于此结构模型,提出了测试用例的设计和选择的方法,从而更好地支持了 Web 应用软件的结构测试。

关键词 Web 应用软件,结构模型,Web 应用软件测试

Research on Structural Testing of Web Applications

LU Xiao-li^{1,2} DONG Yun-wei¹

(College of Computer Science, North West Polytechnical University, Xi'an 710072, China)¹

(College of Public Administration, North West University, Xi'an 710069, China)²

Abstract In order to guarantee Web quality and reliability, people attach more and more importance to Web application testing. Analyzing and understanding Web application, mastering inter control flow and data flow information, good structural testing models and methods can be built so as to carry out structural testing based on coverage rate. In this paper, an structural testing model that is five-level described as function level, function cluster level, object level, object cluster level and application level, was offered. Testing methods of designing and choosing test cases were also introduced so as to support structure testing of Web application.

Keywords Web application, Structural testing model, Web application testing

1 引言

随着 Web 技术的飞速发展,Web 应用软件的种类和数量不断增加,并且已经渗入到社会的各个应用领域。Web 应用软件具有的异构、分布、并发和平台无关的特性,使得对 Web 应用软件的测试要比传统程序的测试更加困难,从而在软件测试领域提出了新的挑战。

为了更好地保证 Web 应用软件的质量,必须对其进行充分的分析和理解,掌握其内部的控制流和数据流信息,实施基于覆盖率的结构性测试^[1]。在 Web 应用中,数据不仅存储在脚本函数和文档中,而且可以通过 http 协议在客户端和服务端传递,这样就带来了新的数据交互。本文对传统的数据流测试方法进行扩展,并将其应用于 Web 应用软件的结构测试中。提出了一种 Web 应用软件结构模型,模型分为 5 级,分别为函数级、函数群级、对象级、对象群级和应用级,且基于此结构模型,提出了测试用例的设计和选择的方法。

本文第 1 节是引言;第 2 节介绍了相关的研究工作;第 3 节描述了结构测试模型和测试用例设计方法;第 4 节是案例分析;最后是结束语。

2 相关研究

Web 应用结构测试的基本思想与传统软件测试的白盒

方法相同,即从变量出发,找出其定义使用链,进而得到关于这个变量的测试路径,进行基于覆盖率的测试。如果在运算语句或输出语句中使用变量,则称为该变量的 C-use(computation use),其主要反映的是数据流信息;如果在谓词语句中使用变量,则称为该变量的 P-use(predicate use),其主要反映的是控制流信息。

目前,结构测试的重要技术——数据流测试技术已经广泛地用于传统程序的测试和面向对象程序的测试。文献[2]基于过程流程图测试单个过程,通过过程内部变量的定义-使用链来选择测试路径。文献[3,4]将数据流测试技巧用于过程之间的交互中,用来描述变量在一个过程中定义而在另一个过程中使用的情况。文献[5]将面向对象类测试分为 3 个级别,分别为 intra-method、inter-method 和 intra-object,其中 intra-method 测试类中的单个方法的数据流;inter-method 测试通过调用关系有交互的多个方法间的数据流;intra-object 测试由不同事件激发的类中方法的不同调用序列的数据流。

在 Web 应用中,DOM(Document Object Model)技术通过将 html 文档中的元素作为带有属性和事件的对象,使得 html 文档能够操作。这就意味着可以把 html 文档中的元素当作变量来使用,并且可以通过文档中嵌入的脚本进行操作,数据不仅存储在脚本函数中,而且存在于文档元素中^[6,7]。另外,Web 应用允许数据通过 http 协议在客户端和服务端

传递,这样就带来了新的数据交互。因此,数据流测试技巧应该被扩展,以用于 Web 应用测试。文献[8,9]将分级数据流测试的思想运用于 Web 应用测试中,但没有考虑文档对象中定义的变量的全局和局部使用时产生的数据流,也没有考虑页面中所嵌入的组件(例如 javabeen)产生的数据流。本文基于上述思想,进一步完善结构模型,提出当测试 html 和 xml 文档、各种组件、跨越客户端页面和服务端页面的 http 请求时,不仅要得到软件组件的变量的定义-使用链,而且必须得到 html 和 xml 文档的变量的定义-使用链,以及跨越客户端页面和服务端页面的变量的定义-使用链。基于这些变量的定义-使用链,可以有效地选择 Web 应用程序的测试路径。

3 Web 应用软件的结构模型和结构测试

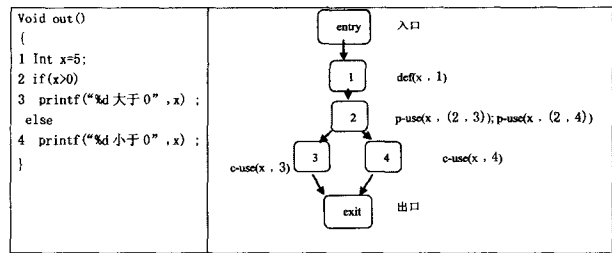
3.1 Web 应用软件的结构模型

Web 应用软件的结构模型共分为 5 级,分别为函数级、函数群级、对象级、对象群级和应用级。对于“intra-object”的情况,需要在函数级、函数群级和对象级寻找变量的定义-使用链;对于“inter-object”的情况,需要在对象群级寻找变量的定义-使用链;对于“inter-client”的情况,需要在应用程序级寻找变量的定义-使用链。

1) 函数级

函数级测试用来测试单一的函数,变量的定义-使用链局限在函数内部,与传统的数据流测试类似。函数级的定义-使用链可以从函数的控制流图(Control Flow Graph,CFG)中得到。CFG 是有向图,节点表示语句块,边表示语句块之间的执行流。例如,如果节点 i 有清楚的定义,节点 j 有变量的使用,则 C-use 使用链 $\langle i, j \rangle$ 存在;如果节点 i 有清楚的定义,边 $\langle j, k \rangle$ 引用了变量,则 P-use 使用链 $\langle i, \langle j, k \rangle \rangle$ 存在,一个测试路径会覆盖一个 C-use 使用链或者 P-use 使用链,如表 1 所列。

表 1 函数级 CFG 例子



对于表 1 左侧的程序,其 CFG 如表右侧的图形所示。对于变量 x ,其定义-使用链为

Def-c-use: $\langle 1, 3 \rangle \langle 1, 4 \rangle$
Def-p-use: $\langle 1, (2, 3) \rangle \langle 1, (2, 4) \rangle$

2) 函数群级

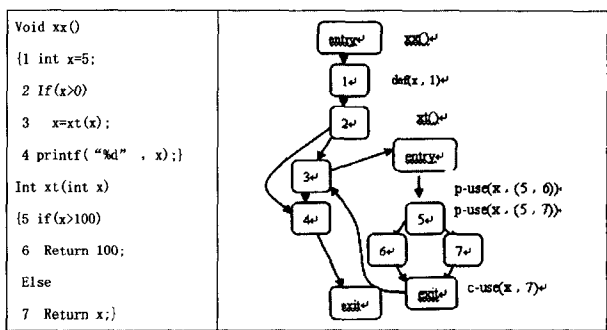
函数群级的测试用来测试对象内部的函数群,变量的定义-使用链涉及到多个函数。一个对象内部的函数群是互相调用的函数集合。因此,利用调用关系,对象内部的函数被划分为几个函数群,每个函数群的定义-使用链都可以从跨过程控制流图 ICFG(interprocedural control flow graph)得到,函数群级的测试路径基于跨过程的变量的定义-使用链得到。

对于表 2 左侧的程序,其 ICFG 如表右侧的图形所示。对于变量 x ,其函数群级的定义-使用链为

Def-c-use: $\langle 1, 7 \rangle$

Def-p-use: $\langle 1, (5, 6) \rangle \langle 1, (5, 7) \rangle$

表 2 函数群级 ICFG 例子



3) 对象级

对象级测试用来测试对象内部不同的函数调用序列,变量的定义-使用链可以通过不同的调用序列改变。在 Web 应用中,由于全局变量,例如 html/xml 文档元素,可以被每个脚本本操作,不同的事件调用序列会产生变量不同的定义-使用链,因此计算不同的函数调用序列的定义-使用链是必要的。为了计算对象内部不同函数激发序列产生的定义-使用链,引入了对象控制流图(Object Control Flow Graph, OCFG)。OCFG 是对象内部的 ICFG 或 CFG 的连接。在 OCFG 中,增加 loop 和 out 节点,loop 节点与所有函数群的 entry 节点连接,函数群的所有 exit 节点连接到 out 节点,从 out 节点到 loop 节点加一个边,以便对于任意的函数调用序列,在连接的 ICFG 中都存在一条路径。

4) 对象群级

对象群级测试用来测试对象群,变量的定义-使用链在群中是跨对象的。一个对象群是通过消息传递关联起来的对象的集合。在 Web 应用中,当用户点击了超链接或者提交了表单,数据就从客户端页面传递到服务端页面。在此,引入复合控制流图(CompositeControl Flow Graph, CCFG),计算跨页面的定义-使用链,CCFG 是交互的页面间的 CFG 或 ICFG 的连接。在 CCFG 中,对于客户端页面的函数群,在其 exit 节点有一条边连接到相应的服务端页面的 ICFG 的 entry 节点。这个边表示 http 请求时,存在一个从客户端页面的 ICFG 到服务端页面的 ICFG 的数据流。例如,图 1 表示 login. html 和 verify. asp 的对象群级合成控制流图,这两个页面通过 http 请求传递了变量 username 和 password。

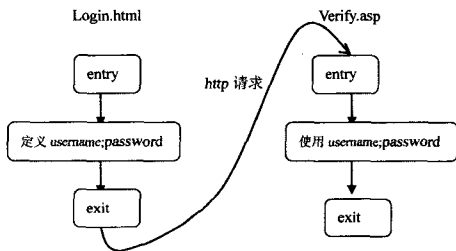


图 1 CCFG 示例

5) 应用级

应用级测试主要用来测试应用程序范围内的变量,这些变量的定义-使用链跨不同的客户。在 Web 应用中,所有访问 Web 应用的客户都可以共享一些变量,这些共享变量成为应用级变量,例如访问站点的客户计数变量。

3.2 基于结构模型生成测试用例

Web 应用的结构测试基于结构模型设计测试用例,选择的测试路径覆盖所有的定义-使用链,测试用例主要通过寻找每个级别的变量的定义-使用链来产生。函数级的定义-使用链可以从函数的控制流图(CFG)中得到;函数群级的变量的定义-使用链涉及到多个函数,每个函数群的定义-使用链可以从 ICFG(interprocedural control flow graph)得到;对象级变量的定义-使用链可以从对象控制流图 OCFG 得到;对象群级的变量的定义-使用链是跨对象的,可以从合成控制流图 CCFG(Composite Control Flow Graph)得到;应用级变量的定义-使用链跨不同的客户,是访问 Web 应用的客户共享的变量。

4 案例分析

下面给出一个例子来说明如何构建结构模型,并基于结构模型选择测试用例,从而支持结构测试。例如,某个拍卖网站主要包含以下页面:拍卖选项选择页面 cbid. html 和服务端页面 bid. asp。用户可从 cbid. html 选择一项;然后输入 userId,并输入略高于当前最小的拍卖价的价格进行提交;请求被 bid. asp 处理后,如果竞价成功,则返回确认信息,否则返回错误信息^[8],如图 2 所示。

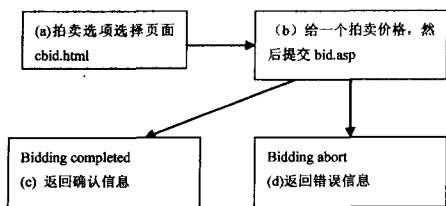


图 2 拍卖应用示意图

cbid. html 的代码如下:

```
1 <HTML>
2 <HEAD><TITLE>eBid Example</TITLE>
3 <SCRIPT LANGUAGE="javascript" TYPE="text/javascript">
4 <! --review and check user's bid
5 function review_bid(){
6   var currentBid=parseFloat(document.all.ptag.innerText);
7   If(parseFloat(document.all.bid.value)>currentBid){
8     document.all.htag.innerText= document.all.userId.value+"
     your maximum bid is"
9     document.all.ptag.innerText= document.all.bid.value;
10    document.all.bid.value=""
11   Else
12     {Error_handler(currentBid);
13     } //display error message
14   Function Error_handler(currentBid){
15     Alert("you need to place a bid more than"+currentBid);
16   } //check if the field is empty
17   Function validate_form(){
18     Var validity=true;
19     If(document.frmMain.userId.value==""){
20       Alert("you must enter your user ID")
21       Validity=false;
22     Else if(document.frmMain.bid.value==""){
23       { Alert("you must enter your maxmimu bid")
```

```
24     Validity=false;}
25     Return validity;
26   }
27 </script></head><body>
28 <h3 id=htag>current maximum bid is </h3>
29 <p id=ptag>20.00</p>
30 <p> of bids;<font size="2" color="Acc3333">8</p></font>
31 <form name="frmMain" action="Bid.asp" method="post" on-
    Submit="return validate_form()">
32   Enter your userID<INPUT type="text" name="userId" size=
    25><br>
33   Enter your maximum bid<INPUT type="text" name="bid"
    size=5><br>
34   <INPUT type="button" onClick="review_bid()" value="Re-
    viewBid">
35   <INPUT type="Submit" value="placeBid">
36 </form></body></html>
```

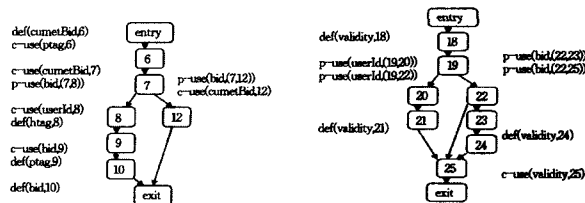
bid. asp 的代码如下:

```
37 <%@ language="vbscript">
38 <html>
39 <head>
40 <TITLE>eBid result</TITLE>
41 </head>
42 <body>
43 <h3>bid results</h3>
44 <%
45   Dim userid,bid,result,TransBid
46   Set TransBid=CreatObject("BidAgent")//retrive data of userid
   and bid
47   Userid=request.form("userId")
48   Bid=request.form("bid")
49   Result= TransBid.placeBid(userid,bid)
50   If result==0 then
51     Response.write(userid & ",your bid is completed")
52     Application("bid#")= Application("bid#")+1
53   Else
54     Response.write("userId or password is error")
55   Endif
56   %>
57 <body>
58 </html>
```

下边说明 5 个级别的结构测试模型。

(1) 函数级

1) 在 cbid. html 中,包括两个函数:review_bid() 和 validate_form(),这两个函数的控制流图如图 3 所示。



(a) review_bid() 的控制流图

(b) validate_form() 的控制流图

图 3 控制流图

在 cbid. html 中,文档元素包括全局变量 ptage, bid, htag 和 userid,其中 ptage 在第 29 行静态地赋值为 \$ 20.00,userid

在第 32 行依赖用户的输入赋值, bid 在第 33 行依赖用户的输入赋值, htag 在第 28 行被使用, 这些变量在脚本函数 review_bid() 中被操作; userId 和 bid 还在 validate_form() 中被操作。因此, 考虑函数级变量的定义-使用链时, 还要考虑这些变量的定义和使用。变量 validity 只在 validate_form() 中定义和使用。函数级变量的定义-使用链如表 3 所列。

表 3 函数级变量的定义-使用链

变量	定义使用链
bid	<33,(7,8)><33,(7,12)><33,9><33,(22,23)><33,(22,25)>
currentBid	<6,7><6,12>
htag	<8,28>
ptag	<29,6><9,29>
userid	<32,8><32,(19,20)><32,(19,22)>
validity	<18,25><21,25><24,25>

2) 在 bid.asp 中嵌入了代码, 共定义和使用了 3 个变量, 分别为 userId, bid 和 result。控制流程图如表 4 左侧的图所示; 函数级变量的定义-使用链如表右侧所示。

表 4 bid.asp 的控制流程图和定义-使用链

	userId	<47, 49><47, 51>
	bid	<48, 49>
	result	<49, (50, 51)> <49, (50, 54)>

(2) 函数群级

在 cbid.html 中, 变量 currentBid 在 review_bid() 中被定义、在 error_handler() 中被使用, 因此函数群级数据流测试要分析 currentBid。currentBid 的定义-使用链为 <6, 15>。函数群级的 ICFG 如图 4 所示。

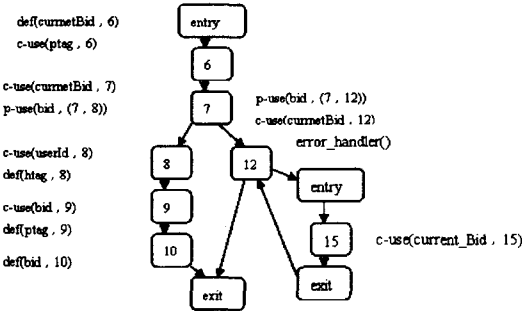


图 4 函数群级的 ICFG

(3) 对象级

对象级的定义-使用链可以从不同的调用序列中得到, 对象的控制流程图如图 5 所示。若用户先用 review 调用了 review_bid(), 然后 submit 调用了 validate_form(), 则存在变量 bid 的 Def-p-use 链 <10, <22, 23>>, <10, <22, 25>>; 但是, 若用户调用了 review_bid() 两次, 则变量 bid 的 Def-p-use 链变成 <10, <7, 8>>, <10, <7, 12>>。

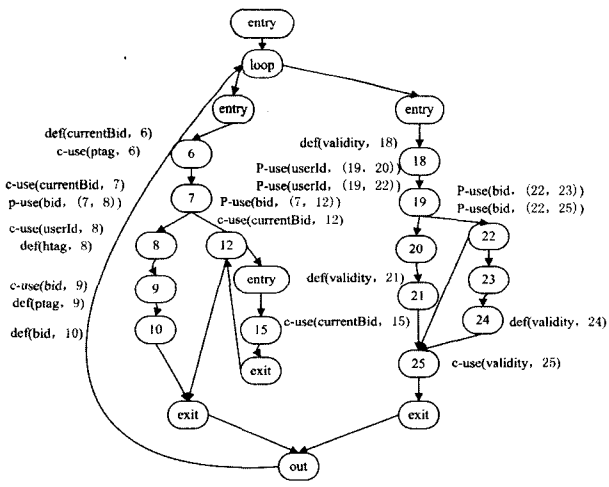


图 5 对象的控制流图

(4) 对象群级

对象群级的定义-使用链可以从集成的 ICFG 中得到, 涉及到 cbid.html 和 bid.asp。当 cbid.html 发送 http 请求时, 由于 userId 在第 32 行依赖用户的输入赋值, bid 在第 33 行依赖用户的输入赋值, 当它们被提交时, 变量 bid 在对象群级的定义-使用链为 <33, 48>, 变量 userId 在对象群级的定义-使用链为 <32, 47>。此外, 由于用户提交前会点击 review 按钮检查竞价价格, 提交时会从 review_bid() 到 validate_form() 的间接数据流, 故还存在变量 bid 在对象群级的定义-使用链为 <10, 48>。

(5) 应用级

在 bid.asp 中, 有一个应用级变量 bid#, 跟踪一个项目的竞价数目。bid# 的定义-使用链为 <52A, 52B>, 跨了两个并发的客户 A 和 B。

结束语 为了保证日益复杂的 Web 应用软件的质量和可靠性, 必须对其进行充分的分析和理解, 掌握其内部的控制流和数据流信息, 实施基于覆盖率的结构测试。Web 应用包含了多种数据操作, 数据不仅存储在脚本函数和文档中, 而且可以通过 http 协议在客户端和服务端传递。本文将传统的数据流测试方法扩展后应用于 Web 结构测试中, 提出了一种 Web 应用软件结构模型。结构模型分为 5 级, 分别为函数级、函数群级、对象级、对象群级和应用级。基于此结构模型, 提出了测试用例的设计和选择的方法。未来将重点研究 Web 应用软件中出现的间接数据流问题, 例如通过持续 XML 数据文件所带来的间接数据流。

参考文献

[1] 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

[2] Rapps S, Weyuker E J. Selecting Software Test Data Using Data Flow Information[J]. IEEE Trans. Software Engineering, 1985, SE-11(4): 367-375

[3] Harrold M J, Soffa M L. Interprocedural Data Flow Testing[C]// Proceedings of the Third Testing, Analysis, and Verification Symposium. Dec. 1989: 158-167

体将样本案例表示成本体的形式,并计算与案例库中的 k 个案例集合中心的距离,把该案例添加到相似度最大的一个案例集合中。当终端用户需要检索时,同样需要将用户提交的任务案例表示成本体,并且与案例库中的标准案例的案例集合中心进行相似度计算,并在返回的最相似度案例集合中继续进行单个案例匹配。匹配的相似度计算方法同上节中的一样涉及到以概念为中心的概念相似度、属性相似度、实例相似度和关系相似度。最终将案例按照相似度从大到小的顺序返回给用户。案例库中的标准案例在数据库表中全部使用本体来表示,其中 *CaseCollection* 表的数据就是 K 个聚类案例集合的详细信息,包括游离系数 d_para 、凝聚系数 n_para 。

5 实验结果分析

本系统在内存为 1G 的戴尔 Inspiron6400 笔记本上运行,数据库服务器安装在 Dell 专用服务器上,并且通过局域网链接,采用的是 Eclipse3.5 的 Java 开发环境。我们采用公司里的营销案例库作为实验的样本案例库,将其中的 30% 作为测试案例库。检索图如图 4 所示。由于将系统设置中的相似度限制为 10%,因此检索出来的案例与提交的范例案例相似度大于 0.10。

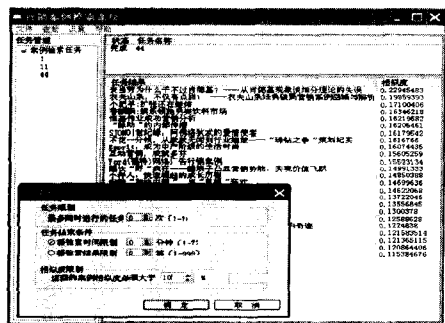


图 4 系统检索界面图

文本检索的两个评测因子通常为查准率 p 和查全率 r , 见式(3)、式(4) [7]。

$$p = \frac{|\{rc\} \cap \{cr\}|}{|\{cr\}|} \tag{3}$$

$$r = \frac{|\{rc\} \cap \{cr\}|}{|\{rc\}|} \tag{4}$$

式中, $\{rc\}$ 为整个测试集中实际与某案例文档相匹配的文档的集合, $\{cr\}$ 为检索结果集。 p 表示检索结果中实际与某案例文档相匹配的个数与整个测试集中实际与某案例文档相匹配的个数的比率,也即匹配的准确率(precision)。 r 表示检索结果中实际与某案例文档相匹配的个数与检索结果中的案例

文档个数的比率,也即查全率(recall)。本案例检索系统使用及不使用 K -means 聚类算法的检索效果如表 1 所列,其中 R 表示系统响应时间,其单位为 s(秒), $Recall$ 表示查全率, $Prec.$ 为准确率。

表 1 实验结果数据对比

	R. (s)	Recall	Prec.	R. (s)	Recall	Prec.
No	25.486	97.313	87.533	54.155	96.654	84.341
Yes	3.237	92.527	89.380	5.176	92.618	86.127
Scale	1000 级别			10000 级别		

从表 1 中可以看出,基于粗糙集的 K -means 在营销案例检索系统中以系统的检索响应时间、检索的准确率和查全率为检索目标,能够大大提高检索的效率,但会对查全率有一定的影响。不过,相比较于响应时间上的突破,这个代价还是值得的。

结束语 本文主要探讨了聚类方法在案例检索中的应用研究,详细地介绍了将聚类方法应用于案例检索系统中,并作了有益的探索,为案例库的检索提供了一条有效的途径。基于粗糙集的 K -means 在营销案例检索系统中以系统的检索响应时间、检索的准确率和查全率为检索目标,能够大大提升用户的满意度。但是 K 的取值和初始点的确定还是个难点,若取值不当反而会影响聚类的效果。因此,以后的研究主要集中在如何确定 K 值和初始点的选定问题上。

参考文献

[1] Pawlak Z. Rough set theory and its application to data analysis [J]. Cybernetics and Systems, 1998, 29(9): 661-668

[2] Raymond T N, Han Jiawei. CLARANS: A Method for Clustering Objects for Spatial Data Mining [J]. IEEE Transaction on Knowledge and Data Engineering, 2002, 14(5): 1003-1016

[3] Pawan L. Applications of Rough Set Based K-Means, Kohonen SOM, GA Clustering [J]. Transactions on Rough Sets VII, 2007: 120-139

[4] Dubois D, Gottwald S, Hajek P, et al. Terminological Difficulties in Fuzzy Set Theory-The Case of Intuitionistic Fuzzy Sets [J]. Fuzzy Sets and Systems, 2005, 156: 485-491

[5] 董振东. 知网主页 [OL]. <http://www.keenage.com>, 1999: 1-15

[6] Wang Dong, Xiang Yang, Zou Guobing, et al. Research on Ontology-Based Case Indexing in CBR [C]// International Conference on Artificial Intelligence and Computational Intelligence. US, Springer, 2009: 238-242

[7] Wikipedia Home page [OL]. http://en.wikipedia.org/wiki/Precision_and_recall, 2000

(上接第 113 页)

[4] Pande H, Landi W, Ryder B G. Interprocedural Def-Use Associations for C Systems with Single Level Pointers [J]. IEEE Trans. Software Engineering, 1994, 20(5): 385-403

[5] Harrold M J, Rothermel G. Performing Data Flow Testing on Classes [C]// Second ACM SIGSOFT Symposium on the Foundations of Software Engineering. Dec. 1994: 54-163

[6] Yang J, Huang J, Wang F, et al. An Objected-oriented Architec-

ture Supporting Web Application Testing [C]// Proc. of Computer Software and Applications Conference, 1999

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

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

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