第 25 卷第 3 期 华 东 交 通 大 学 学 报 Vol. 25 No. 3
2008 年 6 月 Journal of East China Jiaotong University Jun. ,2008

文章编号: 1005 - 0523(2008) 03 - 0069 - 05

基于 XML 的单元自测应用的设计实现

丁振凡,李卓群

(华东交通大学 信息工程学院,江西 南昌 330013)

摘要: 给网络课件每个学习单元安排一个单元自测,不仅增强课件的交互性,也可让学生通过测试更好地理解知识内容. 该应用采用 XML 表示和存储试卷,利用客户端 Javascript 脚本访问 XML 并结合 DHTML 技术实现试卷的显示,在显示处理中利用 DOM 和 XSL 技术实现 XML 试卷内容的访问与变换处理. 该应用可以 Web 方式访问,也可按文件访问方式使用.

关 键 词: 单元自测; XML; Javascript; XSL; DOM

中图分类号: TP391 **文献标识码:** A

教学测试是评价学生知识掌握的重要手段. 在课件中给每个学习单元安排一个教学测试,不仅增强课件的交互性,更重要的是可让学生通过测试更好地掌握和理解知识内容. 英语学习等单机版的教学课件中有不少提供了引人入趣的自测功能. 但在网络课件中此类功能却少见. 常见的 Web 环境下教学测试系统,通常采用数据库或 XML 结合服务方和客户方的脚本技术实现[1],必须在 Web 环境下使用,往往依赖服务器环境. 本文介绍的单元自测应用是一个完全在客户方浏览器上的交互应用,不存在对服务器平台的依赖性,具有很好的环境适应性. 各单元的试卷分别存储在 XML 文件中,由 Javascript 根据当前学习的单元,装载相应的试卷. 然后,采用 XML DOM 技术和 XSL 变换技术访问 XML 试卷内容,利用 DHTML 技术生成相应的试卷解答界面. 依据事件代码实现解答登记和评分处理,并将解答对比显示给学生. 试卷解答操作界面如图 1 所示.

图 1 单元测试做题界面

1 测试试卷的 XML 表示

为节省篇幅,这里给出了三类试题,每类试题给出一道题作为样例. 整个试卷的根结点为 paper,各类试题的相关标记的含义见表 1 所示.

以下为 XML 试卷文件样例.

```
<? xml version = "1.0" encoding = "gb2312"? >
< paper examtime = "100" >
< singlechoice >        <! —单选题 - - >
< score > 1 < /score >
< shiti >
< content >
如何定义一个不能有子类的类 Key?
A) class Key { }
B) abstract final class Key { }
C) native class Key { }
D) class Key { final; }
E) final class Key { }
```

收稿日期: 2008 - 3 - 10

作者简介: 丁振凡(1965 -) ,男,江西丰城人,研究方向为电子商务

```
</content>
<answer>E</answer>
</shiti>
<shiti>…</shiti> <!--下一道试题-->
</singlechoice> <multichoice> <!—多选题-->
<score>1</score>
<timu>
<content>
```

下列哪些是 java 中合法的修饰符?

A. private

B. public

C. protected

D. protect

E. friend

```
</content>
<answer>ABC</answer>
</timu>
<timu>...</timu> <!--下一道试题-->
</multichoice>
<integrity>  <!—完型填空题-->
<score>2</score>
<material>
<mycontent>public class Java-3
{   int x,y; //点的坐标
  public Java-3() {}
  public Java-3(int x,int y) { <wk>this.x=x; this.y=y; </wk>}
  public Java-3(Java-3 p) { <wk>x=p.x; y=p.y;
</wk>}
} </mycontent>
<mycontent>…</mycontent> <!--下一段落-->
</material>
<material>…</material> <!--下一题-->
</integrity>
</paper>
```

说明:同一类试题的各道题的分数相同,所以 Score 标记作为试题类的子结点,而没有安排作为每道试题的子结点. 填空题每道试题为一份阅读材料(material),其中包含若干段落(mycontent),在每段中安排一些空(wk). 各空的标准答案直接在 wk 标记中.

表 1 试卷中 XML 标记的含义

试题类型	标记	含义
单选题	singlechoice	单选题类的根结点
	score	每道试题的分数
	shiti	一道试题的根结点
	content	本道试题内容
	answer	本道试题标准答案
多选题	multichoice	多选题类的根结点
	score	每道试题的分数
	timu	一道试题的根结点
	content	本道试题内容
	answer	本道试题标准答案
完型填空题	integrity	填空题类的根结点
	score	每空分数
	material	阅读材料
	mycontent	材料中一段内容
	wk	试题内容中一个空

2 考试解答界面生成及显示处理

2.1 试卷的装载、

定义一个函数 getData,它将在页面加载时由 OnLoad 事件触发执行,该函数将完成试卷 XML 文件的加载,并通过对 XML 文件的分析处理生成试卷解答界面的 HTML 代码,最后用 DHTML 技术将解答界面的 HTML 代码显示在页面中.

```
function getData(){
    xmldoc = new ActiveXObject("Microsoft.XMLDOM");
    xmldoc.async = false;
    url = "exam" + zh + ".xml"; //zh 为当前学习的章号
    xmldoc.load(url); //装载 XML 试卷
    root = xmldoc.documentElement; //根结点
    …//各类试题的解答界面生成代码
}
```

2.2 各类试题解答界面的生成处理

通过字符串的拼接得到各类试题解答界面的 HTML 代码. 在各类试题的解答界面生成处理中,一方面要显示试题内容,另一方面要生成相应的解答控件,同时要解决解答控件的命名和相应的事件处理定义问题.

1) 单选题解答界面的生成处理代码

```
res = "一、单选题</font><font color='green'><b>(每小题" + root.selectSingleNode("singlechoice/score").text + "分)</b></font><td>"
```

```
items = root. getElementsByTagName( "singlechoice/shiti/content")
ct = 0 //记录试题序号
for ( i = 0; i < items. length; i + + ) {
  ct = ct + 1;
  res = res + "< tr > < td colspan = '3' > < pre >
< font face = '黑体' color = 'blue' > < b > " + ct + "
< /b > < /font > . " + items( i) . text + "< /pre >
< /td >
< /tr > < tr > < td width = '10%' > < /td >
< td colspan = '2' width = '90%' > "
  res = res + "< input type = 'radio' value = 'A' name
= 'x" + ct + "' onclick = 'log( this, " + ct + ") ' >
A    "
  …… //其他选项类似
  res = res + "< /td > < /tr > "
}
```

说明: 在单选题的解答界面生成处理中有两个要点: ①是控件的命名,这里采用 "x" 作为前缀后跟试题序号的办法,注意同一题的各单选按钮名称一样,如此才能达到单选的目的. ② 事件的处理,如何将用户选择的解答通过事件操作记录下来,以便评分处理. 这里采用了传递试题序号和代表按钮控件对象本身的变量 this 的办法实现解答选择信息的传递,试题序号知道做的是哪道题,而 this 参数则可以知道点击了哪个选项,在事件处理函数中可以根据该参数获取选项的值.

2) **多选题解答界面的生成处理代码**

多选题解答界面处理与单选题的处理类似,只是生成的控件用复选框,且同一题的各个复选框不同名,前缀分别用 m1 - ,m2 - ,…表示. 解答处理函数传递的信息要 3 个参数,第一个参数是 this,代表正在操作的复选框对象,第 2 个参数是选项的序号,以便登记时知道选择了哪个选项,第 3 个参数是试题序号. 以下为第 1 个选项的生成代码:

```
< input type = 'checkbox' value = 'A' name = 'm1 - " +
ct + "' onclick = 'logmultichoice( this,1, " + ct + ") ' >
```

3) **填空题解答界面的生成处理代码**

填空题将针对试题中的空的位置生成相应的文本框供用户填写解答,处理上涉及到空的查找与转换,这里采用 XSL 变换进行处理[2].

```
res = res + "< p > < font color = 'blue' > < b > 三、多填空题 < /font > < font color = 'green' > ( 每空 " +
root. selectSingleNode( "integrity/score") . text + "分)
< /b > < /font > < /p > "
root = root. selectSingleNode( "integrity") //选取填空类试题的根结点
xsldoc = new ActiveXObject ( "Microsoft. XMLDOM") ;
xsldoc. async = false;
xsldoc. load( "exam2. xsl") ; //装载 XSL 程序
x = root. transformNode( xsldoc) ; //对整个填空类试题进行变换并返回变换结果
res = res + x; //将结果拼接到 HTML 显示串中
```

以下为相应的 XSL 程序(exam2. xsl)

```
< ? xml version = "1. 0" encoding = "GB2312" ? >
< xsl: stylesheet xmlns: xsl = "http: //www. w3. org/TR/WD - xsl" languange = "JavaScript" >
< ! - - 模板 1 - - >
< xsl: template match = "integrity" >
< xsl: eval > var st = 0; "" < /xsl: eval >
< xsl: eval > var ct = 0; "" < /xsl: eval >
< xsl: for - each select = "material" >
< tr > < td colspan = '3' >
< xsl: apply - templates select = "mycontent" / >
< /td > < /tr >
< /xsl: for - each >
< /xsl: template >
< ! - - 模板 2 - - >
< xsl: template match = "mycontent" >
< pre > < b > < font color = '#800000' > < xsl: eval > st = st + 1; '试题" + st + ':" < /xsl: eval > < /font >
< /b >
< xsl: apply - templates select = "wk | br | text( ) " / >
< /pre >
< /xsl: template >
< ! - - 模板 3 - - >
< xsl: template match = "wk" > " < u > < b > < xsl: eval > ct = ct + 1; ( " + ct + ") " < /xsl: eval > < /b > < /u >
< input type = "text" size = "16" >
  < xsl: attribute name = "name" > w < xsl: eval > ct < /xsl: eval >
  < /xsl: attribute >
  < xsl: attribute name = "onchange" > logkong( this, < xsl: eval > ct < /xsl: eval > )
  < /xsl: attribute >
< /input >
```

```
</xsl:template>
<!-- 模板4 -->
<xsl:template match="br"><br/></xsl:template>
<!-- 模板5 -->
<xsl:template match="text()"><xsl:value-of/></xsl:template>
</xsl:stylesheet>
```

说明: 这部分的编程有一定难度,为了实现将填空题中的空用特殊的输入框显示,程序中定义了共5个模板: ① 模板1定义了填空类试题的根结点的处理策略,其中定义了两个变量,一个是st代表试题序号,另一个是ct代表空的序号. 模板中对每道填空题(material)通过for each循环进行匹配处理,

图2 填空题的解答界面

**每份阅读内容(mycontent)调用模板2进行处理. ② 模板2首先计算输出试题的序号,然后对其中的内容(包括文本和填空项)分别调用模板3和模板4进行处理. ③ 模板3用于填空项的匹配处理,它包括计算和显示一个填空项的序号,另一个难点是产生输入文本框,文本框中的属性通过xsl:attribute标记进行定义,标记名用"m"作为前缀后跟空的序号,onchange事件属性定义的函数logkong包括两个参数,一是代表文本框对象的this参数,另一个是空的序号. ④ 模板4实现对XML文本中
标记的特殊处理,以保证
标记按换行处理; ⑤模板5用于文本的匹配处理,它只是将文本的内容照原样输出. 填空题变换后的页面显示效果如图2所示.**

2.3 解答界面的显示处理

函数getData的最后一段脚本代码以及页面内容定义代码如下,脚本中利用DHTML技术将生成的显示字符串(res)赋值给页面中定义的层(me).

```
res = res + "<p>  <input type='button' value='交 卷' onclick='checkresult()'>
</p>"
me.innerHTML = res; //将拼接好的解答界面显示在名字为me的层中
}
</script>
<body onload="getData()">
<div id="bt"></div>
<!-- 显示试卷标题的层 -->
<div id="me"></div>
<!-- 显示试题解答界面的层 -->
</BODY>
```

说明: 试卷标题层的显示在这里未介绍,它只是将第几单元测试的文字显示出来,单元的值,取决于用户章节导航中的选择了哪章学习,需要通过信息传递得到.

3 解答记录、交卷评分及答案对比的显示

3.1 通过数组记录解答

用户通过页面上的各类控件解答试卷,为了便于后面的各类处理,在用户解答的过程中通过事件处理函数将用户的解答记录在各道试题对应的数组元素中,单选和填空题的解答均用一维数组记录,多选用二维数组记录,由于Javascript不能直接定义二维数组,但可通过定义数组的数组实现.

```
var arr = new Array(100); //单选
var multich = new Array(50); //多选
for (k = 0; k < arr.length; k++)
multich[k] = new Array(5); //多选的5个选项
var kong = new Array(30); //填空题
```

3.2 记录解答的事件处理函数

用户答题时通过页面控件定义的事件触发执行相应的函数处理. 单选和多选题的记录分别依托选项按钮和复选框的onclick事件,而填空题的记录依托文本框的onchange事件. 这些事件处理函数将用户的解答登记到数组中.

(1) 单选解答记录函数

```
function log(m,i) {
  arr[i] = m.value;
}
```

(2) 填空解答记录函数

```
function logkong(m,i) {
```

```
    kong [i] = m. value;
  }
  (3) 多选解答记录函数
  function logmultichoice( m,j,i) {
    //第 i 道题的第 j 个选项
      if ( m. checked)
        multich [i] [j - 1] = m. value;
      else
        multich [i] [j - 1] = "";
  }
```

3.3　交卷处理函数

点击“交卷”按钮将触发执行 checkresult 函数,该函数将访问 XML 文件读取每道试题的标准答案与数组中记录的用户解答进行对比,进而计算得分. 为了将解答对比和得分告诉用户,和前面的生成试卷解答页面类似,这里同样要进行显示内容的拼接处理,要在拼接后的 HTML 文本中显示试题内容、标准答案和用户解答. 如图 3 所示. 计算的得分通过对话框提示给用户.

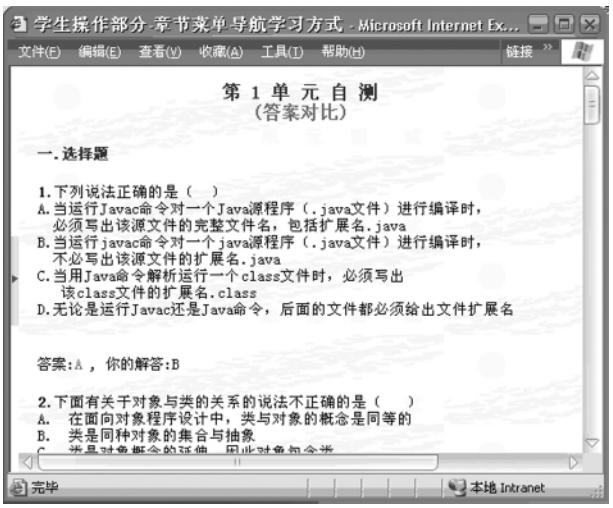

图 3　答案对比显示

以下为函数的部分代码:

```
function checkresult( ) {
score = 0;  //用于累计用户得分
totalscore = 0; //用于累计试题总分值
str = "< center > < strong > < font color = 'red' > (答案对比) < /font > < /strong > "
str = str + "< /center > < br > < table > < td width = 5 > < /td > < td > "
/*  以下为单选题的解答对比处理 * /
s = xmldoc. selectSingleNode ( "//singlechoice/score") ;
xscore = parseInt( s. text) ;  //每题分值
ans = xmldoc. selectNodes( "//singlechoice/shiti") ;
no = 1; //试题序号
mynodes = ans. length;  //单选题数量
str = str + "< font color = 'blue' > < b > 一. 选择题 < /b > < /font > < br > ";
for ( k = 0; k < mynodes; k + + ) {
m = arr [no];
x = ans. item( k) . childNodes. item( 1) . text; //获取试题标准答案
totalscore = totalscore + xscore;
str + = "< pre > < b > " + ( k + 1) + "< /b > . " + ans. item( k) . childNodes. item( 0) . text + "< /pre > < br > ";
if ( m = = x) {
score = score + xscore;
str + = "答案 "+ x + ",你的解答 "+ m + "< br > ";
} else
str + = "答案: < font color = 'red' > "+ x + "< /font > ,你的解答 "+ m + "< br > ";
no = no + 1;
}
… //其他类试题的解答对比显示处理略
str = str + "< /td > < /table > ";
score = Math. floor( score* 100/totalscore) ;
//折算为百分制分数
aler( "考试分数( 折算为百分数) :" + score) ;
//弹出对话框告诉用户得分
me. innerHTML = str; //  显示答案对比
}
```

值得一提的是,填空题的显示处理仍用到 XSL 变换,和前面的处理基本一样,只是对空的处理做了变化,不再显示输入框而是显示空的编号,并加有下划线表示. 以下为相应的模板.

```
< xsl: template match = "wk" >
< u > < b > < font color = 'red' >
< xsl: eval > ct = ct + 1; "(" + ct + ")" < /xsl: eval >
< /font > < /b > < /u >
< /xsl: template >
```

4　结束语

文章介绍的单元自测应用综合体现了客户方编程技术的运用,可广泛应用于网络课件中的教学单元测试. 它既可以通过 Web 形式访问,又可以直接以单机方式使用. 在 Web 访问方式 (下转第 81 页)

http://www.sics.se/~adam/uip/index.php/Documentation. 2004 - 09 - 22.

[3] REALTEK SEMI - CONDUCTOR CO.,LTD. rtl8019 芯片文档 [EB/OL], http://www.ethernut.de/pdf/8019asds.pdf. 2001 - 01 - 03.

Implement of Remote Monitoring System in the Computer Room Based on uIP Protocol Stack

GONG Jin - hong

(School of Electrical and Electronic Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: For the security of real - time monitoring on campus network room, the paper designs the remote monitoring system which observes the real - time state of the environment and server. Alarm information can be sent to the mobile phone through setting a certain threshold. Based on the opening source uIP stack, Web Server module has been developed, which can send the monitoring information to internet browser, and deliver the alarm information to the mobile telephone by GSM communication module.

Key words: uIP; Web Server; GSM

(责任编辑: 周尚超)

(上接第 73 页) 使用时,该应用可部署在任意 Web 服务器上. 而单机方式下,可直接用浏览器以文件浏览方式打开课件进行操作. 脚本代码用客户端执行的 Javascript 编写. 程序中利用 XML DOM 和 XSL 来实现对 XML 内容的访问处理,利用 DHTML 技术实现动态页面显示处理. 该应用具有通用性,各门课程只要编写自己的 XML 文件即可. 在实际应用平台中是将数据库中的试题抽取自动转换产生 XML 格式的试卷.

参考文献:

[1] 丁振凡,网络考试系统的可定制性设计 [J]. 华东交通大学学报,2003,20(5):52 - 55.

[2] Steven Holener. XML 完全探索 [M]. 师夷工作室,译. 北京: 中国青年出版社,2001: 231 - 238.

Design and Implementation of Unit Self Examination Based on XML

DING Zhen - fan, LI Zhuo - qun

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: The arrangement of unit self examination of each unit in network courseware will not only improve the interaction of courseware, but also make student have a better understanding of knowledge. With the representation and store of paper through XML, the software implements the paper display by using DHTML techniques and javascript code to access XML. In the process of paper display, it uses DOM and XSL techniques for the XML access and transformation. The application can be used by web mode and file access mode.

Key words: unit self examination; XML; Javascript; XSL; DOM

(责任编辑: 周尚超)