基于 Ajax 技术的树型结构目录的实现

邹志辉，杨春金，谭勇
武汉理工大学信息工程学院，武汉（430063）
E-mail：aimshiny528@163.com

摘　要：树型结构是人类社会中一种常见的数据结构，随着互联网的发展，应用也越来越广泛。本文介绍了几种实现树型结构目录的方法，并通过一个简单例子，以 Ajax 技术为基础，实现了树型结构目录。

关键词： 树型；Ajax;xml；数据
中图分类号：TP31

1. 引 言

树型结构是一类应用非常广泛的数据结构。人类社会中宗族的族谱和现代企业的组织形式都是树型结构。在计算机领域中,文件系统中文件的管理结构、存储器管理中的页表、数据库中的索引等也都是树型结构[1]。随着 Internet 的飞速发展，树型结构在浏览器/服务器（Browser/Server，简称 B/S）应用系统的应用也越来越广泛。

2. 树型结构目录的实现方法

实现树型结构目录主要有下面几种方法：

2.1 基于 dTree 组件的实现

dTree 是一个免费的 JavaScript 脚本，只需定义有限的几个参数，就可以做出漂亮的树型目录。以下是 dtree 的用法示例：

1）初始化菜单

```
<script type="text/javascript">
  var Tree = new Array;
  Tree[0] = "1|0|Page 1|#";
  Tree[1] = "2|1|Page 1.1|#";
  Tree[2] = "3|1|Page 1.2|#";
  Tree[3] = "4|3|Page 1.2.1|#";
</script>
```

2）调用函数

```
<div class="tree">
  <script type="text/javascript">
    createTree(Tree,1,7);
  </script>
</div>
```

dtree 组件实现树型结构目录的方法函数都在 dtree.js 中，在应用时必须在页面中引用必要的 CSS 和 JS 文件。如下：

```
<link href="dtree.css" type="text/css" rel="stylesheet"/>
<script language="JavaScript" src="dtree.js" type="text/javascript"></script>。
```

2.2 基于 xTree 组件的实现

xTree 是http://webfx.eae.net/公司的基于 JavaScript 的建立树型目录的开源产品，扩展性

强，使用简单。xTree 适合于静态目录树，而另外一种就是 xLoadTree，适合于动态局部加载目录树，利用 Ajax 的思想进行封装的[3]。

1）静态树：

```
<script type="text/javascript">
  var tree = new WebFXTree("测试树");
  for (var i = 0; i < 50; i++){
  tree.add( new WebFXTreeItem("子结点 " + i) ); }
  tree.write();
</script>
```

2）动态树

var tree = new WebFXLoadTree("Hello World", "tree.xml");打开根结点时，即从 tree.xml 文件中得到子树，也可是从服务器返回的 XML 流 tree.write();

而无论是静态树还是动态树，两种树都是从 XML 文件中读到子树结点：

```
var tree = new WebFXTree("Root");
tree.add(new WebFXTreeItem("Tree Item 1"));
tree.add(new WebFXLoadTreeItem("Tree Item 2", "tree.xml"));
tree.add(new WebFXTreeItem("Tree Item 4"));
tree.write();
```

xtree 还提供了两个 JS 文件：xtree2.js，xloadtree2.js，里面包括了 xtree 的工作原理和相关方法及属性。

2.3 基于 Ajax 组件的实现

Ajax 是 Asynchronous JavaScript and XML 的简称，它提供了通用的、动态加载节点的解决方案。

1） 动态加载技术

如果一次性获取完整的先序树，构造成 xml 提供给 JavaScript 解析，数据量越大，消耗的资源越多，客户端响应延迟时间就越长，因此对于大数据量的树，采用动态加载方式，即每次单击“+”图片时，判断是否已加载子节点数据，如果未加载则通过 Ajax 的 XMLHTTP 组件 XMLHTTPRequest 对象异步发送请求[5]。相关 JavaScript 代码如下：

```
/*判断是否已经加载数据，未加载则访问服务器加载数据*/
dhtmlTree.prototype.Loading=function(pObject){
  if(((pObject.XMLload==0)&&(this.XMLsource))&&(!this.XMLloading)){
    pObject.XMLload=1;
this.loadXML(this.XMLsource+getUrlSymbol(this.XMLsource)+"id="+escape(pObject.id
));
  }
}
dtmlXMLObject.prototype.loadXML=function(url){//加载数据
  try {
    this.xmlDoc = new XMLHttpRequest();
    /*通过 GET 方法异步连接到 url 加载数据*/
    this.xmlDoc.open("GET", url,true);//true：异步；false：同步
    this.xmlDoc.send(null);
```

```
        } catch(e){
          this.xmlDoc = new ActiveXObject("Microsoft.XMLHTTP");//使用 IE
          this.xmlDoc.open("GET", url,true);//true：异步；false：同步
          this.xmlDoc.send(null);
        }
        return this.xmlDoc.responseXML;
      }
```

2） 树型结构的构造

从数据库中返回的是有序的先序树，而 XML 是完整的树型结构文档，所以将树型数据构造成预定义的 XML 格式，只需从根节点开始，遍历一遍树，即可将树全部生成[4]。相关 JavaScript 代码如下：

```
dtmlXMLObject.prototype.constructTree=function(){
var node=this.XMLLoader.getXMLTopNode("tree");
var parentId=node.getAttribute("id");
for(var i=0;i＜node.childNodes.length;i++) { //逐个解析 xml 文件的 leaf 节点
if((node.childNodes[i].nodeType==1)&&(node.childNodes[i].tagName == "leaf")){
var name=node.childNodes[i].getAttribute("text");
……
var temp=dhtmlObject.a0Find(parentId);//获取父节点对象
temp.XMLload=1;//已加载
//构造 html 输出节点
dhtmlObject.insertItem(parentId,cId,name,im0,im1,im2,chd);
dhtmlObject.addDragger = this;//设置可拖放的对象
};}
```

3. Ajax 技术的树型结构目录实现

3.1 数据库设计

数据库是用 powerdesigner 设计的，在这只是做演示例子，一共用到三个字段,子节点编号 ChildId,父节点编号 ParentId,节点名称 NodeName，数据库节点表如图 1 所示[2]：

图 1 数据库节点表

子节点，父节点用来记录目录的上下级关系，以便生成目录树。

3.2 程序设计

主要程序用到 AjaxXmlBuilder 辅助类，AjaxXmlBuilder 是一个工具类，它可以包含一些工具方法，这些工具方法根据字符串或者集合来生成满足 AjaxTags 要求的响应[4]。借助于 AjaxXmlBuilder 的帮助，程序开发者无须手动输出<ajax-response>，<response>，<name>

和<value>等标签，只需调用 AjaxXmlBuilder 的响应方法，它就会自动生成这些标签，并将字符串或集合里的对象的值增加到该响应里。AjaxXmlBuilder 对象主要包含如下两个方法。

public AjaxXmlBuilder addItem(java.lang.String name, java.lang.String value)：每调用一次，为响应增加一个 item 节点。其中，第一个参数是 item 节点下 name 节点的值，第二个参数是 item 节点下 value 节点的值。

public addItems(java.util.Collection collection, String nameProperty, String valueProperty)：该方法编辑集合 collection 的值，collection 里的元素是对象，该对象必须包含 nameProperty 和 valueProperty 两个属性；该方法增加 collection 的长度个 item 节点，每个 item 节点的 name 节点的值就是集合元素的 nameProperty 属性的值，而 value 节点的值就是 valueProperty 属性的值。

通过这两个方法，可以非常便捷地生成 AjaxTags 所需格式的响应。

程序如下：

```
public class TreeServlet extends BaseAjaxServlet{
    public String getXmlContent(HttpServletRequest request, HttpServletResponse response)
      throws Exception {
            String node = request.getParameter("node");//得到节点信息
            AjaxTreeXmlBuilder treeBuilder = new AjaxTreeXmlBuilder();  // 调
用 AjaxTreeXmlBuilder 类
            if(node.equals("bh")) {
            treeBuilder.addItem("综合文件", "001", true);//初使化最高节点
            treeBuilder.addItem("监理资料", "002", true);
            treeBuilder.addItem("施工资料", "003", true);
            }
          else
           { Connection con=Db.getConnection();
            Statement stmt=con.createStatement();
            ResultSet rs=null;
            String sql="select ChildId, NodeName from nodetable where ParentId='"+node+'"
";//根据节点信息动态访问数据库
            rs=stmt.executeQuery(sql);
            while(rs.next())
        {
            String s1=rs.getString(1);
            String s2=rs.getString(2);
            treeBuilder.addItem(s2,s1,true,);//从数据库中得到结果集，形成 xml 格式的完整
树型结构文档
        }
            Db.closeRs(rs);
            Db.closeStmt(stmt);
            Db.closeCon(con);
        }
            return treeBuilder.toString();//回调函数}
        }
```

Tree.jsp 页面要引用相关的 css 和 js 资源和编写 ajax 标签：

```
<!-- 在 JSP 页面中引入必需的 JavaScript 库 -->
<script type="text/javascript" src="common/js/ajax/ajaxtags.js"></script>
<script type="text/javascript" src="common/js/ajax/ajaxtags_controls.js"></script>
<script type="text/javascript" src="common/js/ajax/ajaxtags_parser.js"></script>
<link rel="stylesheet" type="text/css" href="common/css/ajaxtags.css" />
<link rel="stylesheet" type="text/css" href="common/css/displaytag.css" />
```

ajax 标签编写：

```
<ajax:tree // ajax 标签
 baseUrl="../TreeServlet" //类文件映射地址
 styleId="bh"
 parameters="node={ajaxParameter}"
 nodeClass="nodeClass"
 expandedClass="expandedNode"
 collapsedClass="collapsedNode"
 treeClass="tree">
</ajax:tree>
```

3.3 页面测试

启动 Tomcat,打开 IE 浏览器，在地址栏输入http://localhost:8080/HY/datamgr/Tree.jsp，回车，树型目录测试结果如图 2 所示：

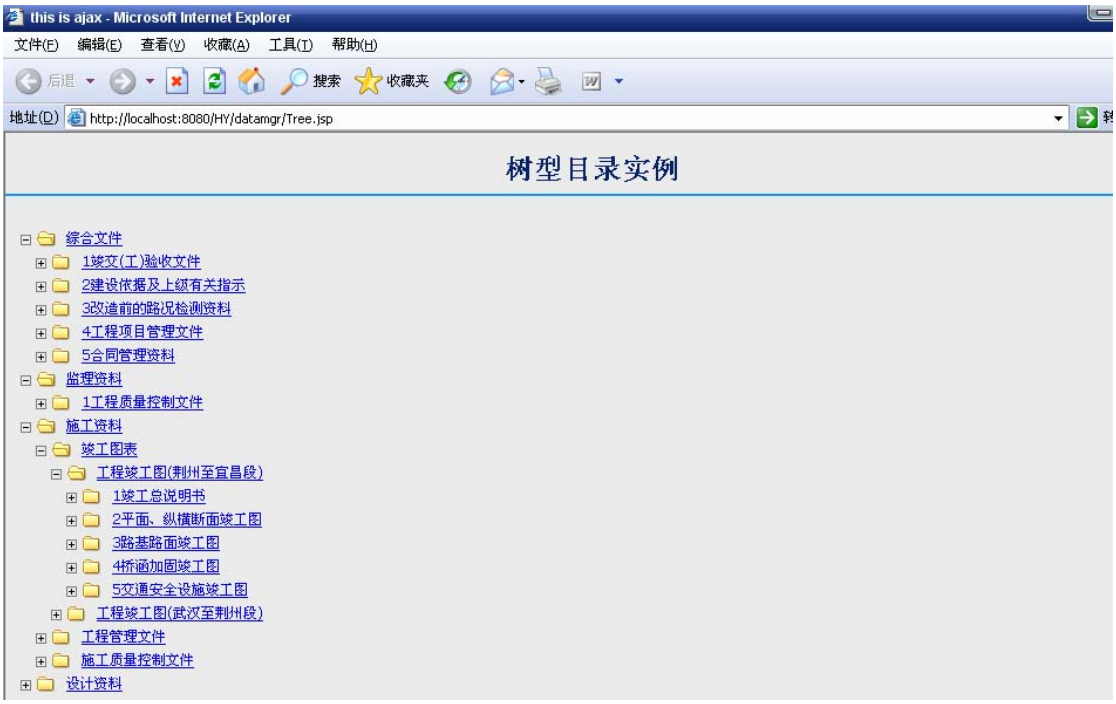

图 2 树型目录测试结果

4. 结束语

本文通过分析编程实现了所需的树型结构目录，整个资料结构清晰地显示出来了。Ajax 作为一种简单实用的 Web 组件，很好地实现了应用信息系统中的树型结构目录，代码编写

量少，对于初学者来说比较好掌握，但是灵活性也受到很多限制。

参考文献

[1] 王辉 ,吴华平.于炳飞. 多级树型目录的动态实现[J].软件导刊,2006.10
[2] 李冠德,陈梦东.Java 中基于 Web 组件的树型目录的设计及实现[J].微机发展,2005.11
[3] 杨光,李龙清,常心坦.基于 Web 实现树型目录两种方式及比较[J],西安科技大学学报,2004.9
[4] (美) Christian Gross 著,李锟 ... [等]译.Ajax 模式与最佳实践[M],北京:电子工业出版社,2007.
[5] Dave Crane, Eric Pascarello, Darren James 著, ajaxcn.org 译.Ajax 实战[M].北京:人民邮电出版社,2006.

Implementation of Tree Type Catalogue Based on Ajax Technology

Zou Zhihui, Yang Chunjin, Tan Yong
Department of Information Engineering, Wuhan University of Technology, Wuhan (430063)

Abstract

Tree type structure is one kind of common data structure in human society , with the development of Internet, also using more and more widely. In this paper, several kinds of ways has been introduced to realizes the tree type structure catalogue, and through a simple application, taken Ajax technology as basis , has realized the tree type structure catalogue .

Keywords: Tree Type; Ajax; xml; Data