

基于网络数据库的 Treeview 设计与实现

刘沙飞 刘武 (武汉华中师范大学物理系 430079)

胡鸿彬 (中科院武汉物理与数学研究所 430071)

摘要: 目录树 (Treeview) 用于显示按照树形结构进行组织的数据, 本文详细介绍了一种基于网络数据库的目录树 (Treeview) 制作技术, 给出了用 PHP, Apache 和 MySQL 实现目录树 (Treeview) 的实例。

关键词: PHP Linux MySQL Apache Treeview 数据库

1 前言

随着互联网的迅速发展, 越来越多的网站采用了类似 Windows 资源管理器中目录树 (Treeview) 的形式来展示站内资源, 网站的浏览者点击目录树 (Treeview) 中的结点, 则该结点展开, 显示下属的子结点。通过目录树 (Treeview), 浏览者不仅能快速搜索到所需的资源, 而且能清楚地认识所有资源的构成结构和层次关系, 以往的目录树 (Treeview) 通常是采用帧 (Frame) 和层 (Layer) 技术, 不仅源代码复杂, 而且往往只能显示固定的目录树 (Treeview) 结构和内容, 站内资源更新时必须重新修改源代码, 本文介绍的是一种基于数据库的目录树 (Treeview) 技术, 该技术将各资源名称、所在层次、级联对应关系等属性储存于后台数据库中, 从而将目录树 (Treeview) 结构的维护转化为对后台数据库的维护, 大大简化了后期的维护工作。

2 系统设计

2.1 系统结构

系统采用用户端浏览器、Web 服务器和数据库服务器 (B/S/DBMS) 三层结构 (如图 1 所示), 与传统的 Client/Sever 两层结构相比, 具有跨平

台、易于维护、信息共享度高、扩展性好、安全性高等优点。

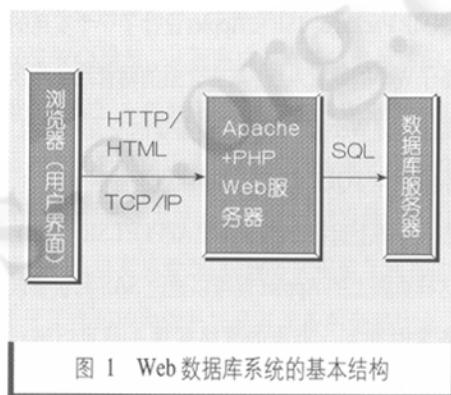


图 1 Web 数据库系统的基本结构

用户通过浏览器向 Web 服务器发出执行请求, Web 服务器读取相应的 PHP 文件, 使用脚本解释引擎对 PHP 文件进行解释执行, 对后台数据库执行查询等 SQL 命令, 然后 Web 服务器再将结果返回用户的浏览器并显示出来。

2.2 网络数据库解决方案

目前动态访问网络数据库的技术有 CGI、ISAPI、NSAPI 和 Script 语言 JSP、ASP、PHP 等, 本系统采用的 PHP 技术与其他数据库开发技术比较起来, 具有以下优势:

(1) 易学易用: PHP 语法结构大部分借用了 C、Java、Perl 的语法框架, 开发周期短, 易于维护。

(2) 跨平台、稳定性和扩展性高: PHP 可以在 Windows、Unix、Linux、Macintosh 等多种平台上运行, 支持 IIS、Apache 等通用 Web 服务器, 拥有良好的扩展性和超常的稳定性, 而且它开放源码, 可以免费使用。

(3) 强大的数据库支持: PHP 最强大也最有代表性的特点在于它内置对很多数据库的支持, 而不再需要特殊扩充, 使得编写基于数据库的网页变得非常简单。

(4) 支持 Internet 的前沿技术: PHP 支持采用 POP3、HTTP、SNMP、NNTP、IMAP 等协议进行网络编程, 还提供对文件上传、密码认证、动态 GIF 生成、内存共享、XML、PDF、GZIP 的直接支持。

2.3 开发平台、数据库和 Web 服务器的选择

由于 PHP 跨平台, 支持多种数据库和 Web 服务器, 用户可以根据自己的实际情况选择相应的开发平台、数据库和 Web 服务器, 本文给出的实例选择了 Linux 系统作为开发平台, 服务器和数据库系统分别选择了 Apache 和 MySQL。

Linux 作为一个优秀的网络操作系统, 具有成本低、性能高、稳定性好和源代码开放等特点, 内核精练而稳定, 而且自带免费的 MySQL 数据库, 是全球使用最广泛的操作系统之一。

尽管 PHP 可以运行在 Microsoft IIS 以及 Netscape Enterprise Server 上, 但与著名的 Apache Web 服务器可以说是最佳结合, Apache Web 服务器是功能强大的免费 HTTP 服务器, 它的特点在于可靠、高效, 并且易于扩展, 所有源代码公开。

MySQL 是一种多用户、多线程的数据库系统, 支持多种操作平台, 执行速度很快, 与 PHP 的配合非常的合适, 而且除供商业用途需要收取少量的费用以外, MySQL 是免费的, 非常适用于学校等中小型事业单位。

3 系统实现

3.1 系统分析

本系统的技术难点在于根据浏览者点击目录树 (Treeview) 中结点的情况动态地调整该结点的

展开和闭合状态。为了在 Server 和 Browser 间传递用户状态数据，我们选择了直接在结点的 URL 上传递变量的 GET 方法，浏览者点击结点链接的同时，也将目录树 (Treeview) 相应的状态数据传递给了服务器，服务器再根据这些数据动态地调整结点的状态，其流程如图 2 所示。

3.2 数据库设计

根据系统功能分析，应当给目录树 (Treeview) 中的每一层结点创建一张表格，结点展开时检索相应层次的表格从而得到该层全部结点的内容，本文给出的实例是远程教育平台中的课件检索

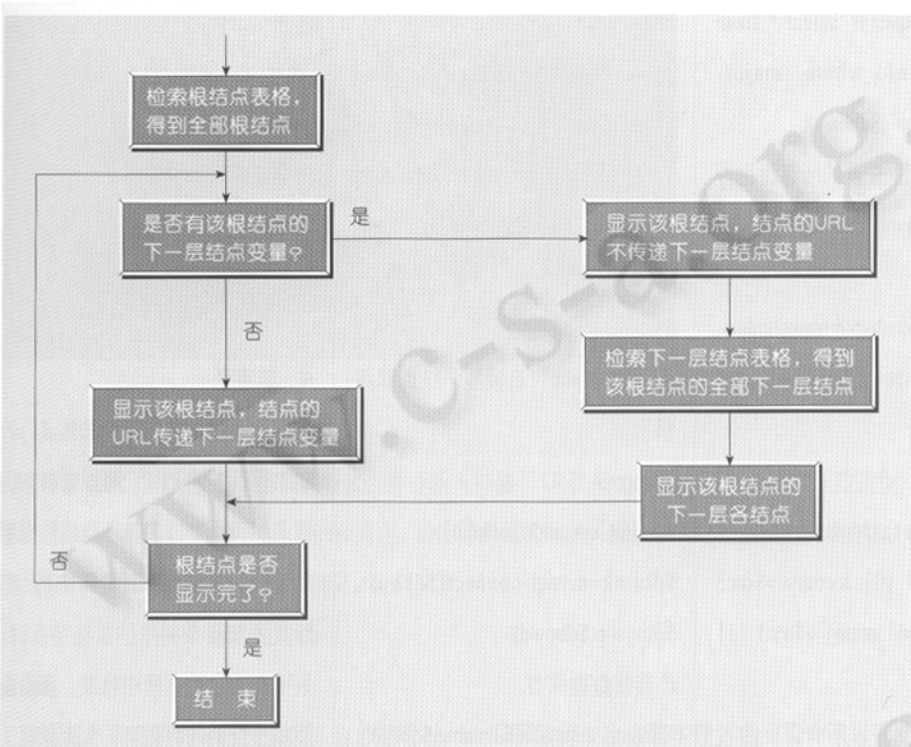


图 2 程序流程图

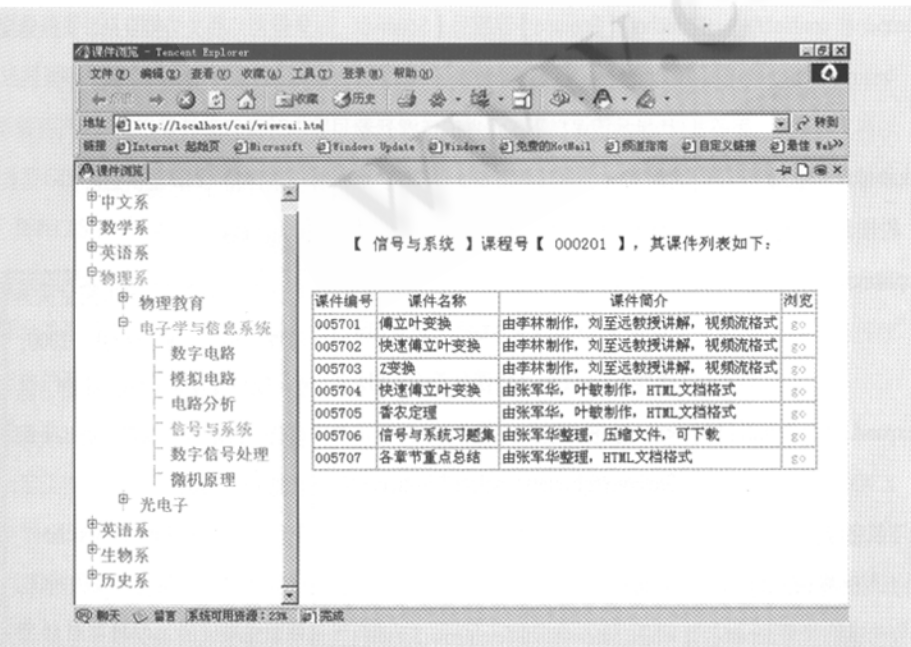


图 3 运行结果

索页面，页面左帧的目录树 (Treeview) 由三层结点 (开设远程教育的各院系，院系下属的专业和该专业开设的课程) 顺次展开组成，页面右帧显示第四层结点 (左帧所选课程下属的课件) 的详细内容，如图 3 所示。因而在 MySQL 数据库中建立四张表：

- (1) 院系信息表 department，存储院系编号，院系名称，院系简介等，主键为院系编号。
- (2) 专业信息表 major，存储专业编号，专业名称，所在院系编号等，主键为专业编号。
- (3) 课程信息表 course-baseinfo，存储课程编号，课程名称，开课专业编号，先行课程等，主键为课程编号。
- (4) 课件信息表 cai-baseinfo，存储课件编号，课件名称，所属课程编号，课件简介，URL 地址等，主键为课件编号。

3.3 程序设计

系统的源代码由以下几个文件组成：

(1) viewcai.htm：利用框架技术将整个页面分为左右两个框架，左边显示树形结构，右边显示浏览者点击结点的详细情况。

```

<html><head><title>课件浏览</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"></head>
<frameset cols="230,517*" frameborder="NO" border="0" framespacing="0" rows="*">
<frame name="left" src="viewtree.php" noresize>
<frame name="main" src="detail.php"></frameset>
<noframes><body>
    本页面使用了帧技术,请使用支持帧页面的浏览器查看
</body></noframes></html>
  
```

(2) common.inc.php：数据库的连接配置文件，包括用户名，密码，数据库名和主机名。

```

<?php
$dbhost="localhost";
  
```

Design and Realization of Treeview System Based on Web Database

```
$dbname="lucy";
$dbsysop="root";
$dbpwd="lucy";?>
```

(3) viewtree.php: 左帧目录树的生成文件。

```
<?php
require("common.inc.php");
$dblink=mysql-connect($dbhost,
$dbsysop,$dbpwd);
/* 连接数据库 */
@mysql-select-db($dbname,$dblink);
$res=mysql-query("select * from
department",$dblink);
/* 检索出所有的院系 */
while($rec=mysql-fetch-row($res)){
if($department==$rec[0]){
/* 如果变量 department 已经存在 */
echo("<a href='\" $PHP-SELF\">");
echo("<no><img
src='\" images/wrap.gif\">$rec [1]<
a><br>");
/* 显示该院系名称, 其 URL 链接地
址为本页面 */
$res2=mysql-query("select * from
major where department-
number=$department",$dblink);
/* 检索出该院系下属的所有专业 */
while($rec2=mysql-fetch-row($res2))
{
if($major==$rec2 [0]){
/* 如果变量 major 存在 */
echo("<no><br>&nbsp;&nbsp;&nbsp;&nbsp;<a
href='\" $PHP-SELF?
department=$department&major=$rec2
[0]\">");
echo("<img
src='\" images/expand.gif\">$rec2[1]
</a ><br>");
/* 显示该院系下属各专业名称, 其
URL 为本页面, 同时将 department
和专业编号分别传递给 变量 depart-
ment 和 major */
```

```
src='\" images/wrap.gif\">$rec2 [1]
</a><br>");
/* 显示该系下属所有专业的名称,
其 URL 链接地址为本页面, 同时继
续传递变量 department */
$res3=mysql-query("select * from
course-baseinfo where major-
number=$major",$dblink);
/* 检索出该专业开设的所有课程 */
while($rec3=mysql-fetch-row($res3))
{
echo("<no>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
&nbsp;&nbsp;&nbsp;&nbsp;<img src='\" images/iterm.
gif\">");
echo("<a
href='\" detail.php?course-
number=$rec3 [0] &course=$rec3
[1]\" target='\" main\">$rec3 [1]
</a><br>");
/* 显示该专业开设所有课程的名
称, 其链接地址为右帧页面, 同时
将课程编号和课程名称 分别传递给
变量 course-number 和 course */
} }
}
}
else {
echo("<no>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
href='\" P H P - S E L F ?
department=$department&major=$rec2
[0]\">");
echo("<img
src='\" images/expand.gif\">$rec2[1]
</a ><br>");
/* 显示该院系下属各专业名称, 其
URL 为本页面, 同时将 department
和专业编号分别传递给 变量 depart-
ment 和 major */
```

```
}}
else {
echo("<a
href='\" $PHP-SELF?
department=$rec [0] \">");
echo("<no><img
src='\" images/expand.gif \">$rec
[1]</a><br>");
/* 显示院系名称, 其 URL 为本页
面, 同时将院系编号传递给变量
department */
}}?>
(4) detail.php: 右帧显示结点详细
情况
<?php
require("common.inc.php");
$dblink=mysql-connect($dbhost,
$dbsysop,$dbpwd);
/* 连接数据库 */
@mysql-select-db($dbname,$dblink);
if($course-number!=""){
/* 如果变量 course-number 存在 */
echo("<[ $course ] 课程号 [ $course-
number ], 其课件列表如下: <br>");
/* 显示该课程下属课件的详细列表
*/
echo("<table border=0 cellspacing=1
cellpadding=0>");
echo("<tr><td>课件编号</td><td>
课件名称</td><td>课件简介</td>
<td>浏览</td></tr>");
$res=mysql-query("select * from cai-
baseinfo where course-
number=$course-n umber",$dblink);
while($rec=mysql-fetch-row($res)){
echo("<tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [0]</td><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [1]</td><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [2]</td><td>&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [3]</td></tr>");
```

```
td>");
/* 显示课件编号 */
echo("<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [1]</td>");
/* 显示课件名称 */
echo("<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [3]</td>");
/* 显示课件简介 */
echo("<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
$rec [2]</td>");
/* 显示课件的链接 */
echo("<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
<a href='\" $rec [4]\">
go</a></td></tr>");
} echo("</table>");}>
程序运行结果如图 3 所示:
```

4 结束语

本系统采用了当今非常流行的动态网页技术 PHP, 将以往静态的目录树页面变成了基于网络数据库的动态目录树页面。需要修改, 删除或添加目录树中的某些结点时, 只要在后台数据库中修改, 删除或添加相应的内容即可, 大大减轻了网络维护的工作量, 而且整个系统运行稳定, 速度快, 非常适用于信息量大、层次结构复杂、更新速度快的各类网站。 ■

参考文献

- 王沫, PHP&MySQL完全实例教程, 电子工业出版社, 2000.
- [美] Leon Atkinson 著, 陈虹译, PHP核心编程, 清华大学出版社, 2000.
- 陈万米, 用 Linux+Apache+Oracle+Php架构电子商务系统, 微型电脑应用, 2000年第16卷第9期(12-14).