

自定义分章图文混编文章存储技术的研究^①

Research on Storing Technology of Long Article with Pictures and Custom Chapters Dividine

林碧英 陈达峰 (华北电力大学 计算机科学与技术系 北京 102206)

摘要: 本文从实际应用出发, 简要介绍了目前常见的文章管理系统的存储特点, 并针对这些文章管理系统所存在的不足之处, 提出了一种基于 ASP.NET 动态网页技术的自定义分章大容量图文混编文章存储技术。该技术主要利用 ASP.NET 开源服务器控件 FreeTextBox 实现文章的在线编辑, 并利用网页标签对文章章节进行分章、定位, 最后利用树形列表动态展示所有文章及章节。目前该技术已成功应用于变电站规程管理系统, 效果良好。

关键词: 文章管理系统 自定义分章 大容量图文混编 动态网页技术 FreeTextBox

1 引言

文章管理系统通常扮演着互联网上信息发布平台的角色, 合法用户只需 IE 浏览器便可登录系统浏览文章, 例如一些新闻网站或者网上电子图书系统等。有的系统使用了常见的文件格式保存文章(如 pdf 格式, doc 格式, 静态网页), 而系统本身只是提供了一定的浏览辅助功能。此类系统的用户浏览文章时需要安装特定的计算机应用软件, 例如需要安装 office 才能正常浏览 doc 文件; 由于文件格式公开, 每一个登录服务器的用户都能直接查看文件内容, 保密程度较低。而另外一些较专业的系统使用了开发公司自定义的文件格式保存文章, 例如《中国电力百科全书》。由于文件格式不常见, 编辑文章过程比较复杂, 系统通常不提供添加新文章和修改原有文章的功能。

综上所述, 现在大部分的文章管理系统所使用的文章存储技术都存在一定的不足之处:

①安全方面

文章以计算机文件的形式保存在服务器中, 非法用户登录计算机后便可轻松盗走信息;

②使用方面

对于使用常见文件格式的系统, 用户需要熟练掌握特定软件的使用才能对文章进行修改和重新分章; 对于许多使用自定义文件格式的系统, 例如一些电子

图书系统, 用户只能浏览文章, 而不能修改, 更不能自定义分章分节。

因此, 需要一种新的存储技术来弥补以上的不足。这种技术能把大型文章保存到数据库中, 并实现图文混编; 用户能够随心所欲地对文章进行分章分节, 浏览时可打开树型结构的章节列表, 选择浏览任意章节。这就是本文要介绍的自定义分章大容量图文混编文章的存储技术。

2 关键技术实现

ASP.NET 是微软推出的全新一代的动态网页实现系统。利用数据库和 ASP.NET 动态网页技术来保存文章信息, 可以很大程度上减少管理人员的工作量, 提高工作效率和系统的安全性。文献[1]介绍了 ASP.NET 技术的一些优点。文献[2]则使用 SQLServer2000 数据库和 APS.NET 技术成功设计了一个文章管理系统, 实现了对文章信息及时、有效的管理。因此本文决定采用数据库和 APS.NET 技术来实现新的存储技术。本文需要解决的问题有: 一般的文章中都会包含图片, 而某些科技类文章中的复杂的公式也需要以图片的形式保存, 因此所使用的文本编辑器要能支持图文混编; 用户把所需的文本和图片添加到文本编辑器后, 可以自由划分章节, 设定章节标题; 保存文章的时候, 文

^① 收稿时间:2008-09-11

本和图片分开保存, 文本存入数据库中, 而图片则存入对应的文件夹; 用户在浏览文章的时候可以通过树形列表选择章节, 使文章展示窗口直接跳转到所选择的章节。

2.1 文章的编辑、存储和显示

FreeTextBox 是一个基于 Internet Explorer 中 MSHTML 技术的 ASP.NET 开源服务器控件。这是一款优秀的自由软件(Free Software), 我们可以轻松地将其嵌入到 Web Forms 中实现 HTML 内容的在线编辑, 适用于新闻发布、博客写作、论坛社区等多种 Web 系统中。FreeTextBox 本身已实现图文混编功能, 用户只需点击 FreeTextBox 上的按钮便可轻松上传图片, 并在 FreeTextBox 自由使用。此外, FreeTextBox 自带了 Word 编辑软件中的大部分常用功能, 并且实现了编辑框中的文本和 html 代码的互相转换, 即用户可以在编辑框的“设计”模式下直接编辑文本, FreeTextBox 本身会自动把所编辑内容转换为对应的 html 代码, 并在编辑框的“html”模式下展示。熟悉 html 语言的用户可以在“html”模式下对文本进行修改, 实现更多的功能。为了防止用户在 html 代码中加入恶意脚本, 服务端需要对 Freetextbox 提交的内容进行检查, 把 html 代码中所有的脚本过滤后再存入数据库。

文本可直接在编辑框中输入, 也可以通过 Windows “复制”“粘贴”功能直接写入 FreeTextBox 的编辑框中; 如果文章包含图片, 则需要把图片单独分离出, 上传至服务器中, 然后在编辑框中相应的位置插入。在编辑框的“设计”模式下, FreeTextBox 已自带图片上传功能, 打开图片库后双击图片, FreeTextBox 会自动把图片插入光标所在位置。用户可在编辑框中调整图片大小, 位置。当切换到“html”模式下, 用户就能看到图片保存的相对地址。图 1 为在 FreeTextBox 编辑框中添加图片的效果, 图 2 为切换到“html”模式后编辑框的内容, 其中 `http://127.0.0.1:750/WebApplication1/images/16036/cowok.jpg` 即图片在服务器中的位置, 16036 为该文章的 ID。

虽然 FreeTextBox 已经实现了图文混编功能, 但

是上传的图片均保存在指定的默认文件夹中。如果需要保存多篇文章, 大量的图片上传到同一个文件夹中显然不方便管理, 因此需要把不同文章的图片分开保存。



图 1 在编辑框插入图片



图 2 切换到 html 模式后的内容

给每篇保存在数据库的文章一个唯一的 ID, 并以 ID 为名新建一个文件夹。在保存文章的时候, 把默认文件夹里的图片全部转移到新文件夹中, 并且把所要保存的 html 代码里的默认文件夹名字替换成新文件夹名字, 以保证图片相对地址的正确; 而在修改文章的时候, 只需把 FreeTextBox 保存图片的默认文件夹改为所改文章对应的图片文件夹, 这样打开图片库时用户便可浏览该文章所对应的图片。

本文使用 Access 数据库的大二进制类型字段保存文本所转换成的 html 代码, 该类型字段可保存 1G 以上长度的二进制文件, 基本满足了一般大型书刊的

保存要求[3]。保存的时候，html 代码会自动转化为二进制文本保存在字段中；读取的时候，需要使用转化函数把从数据库中读出的大二进制类型字段的二进制内容转化成保存前的 html 代码，C#所使用的转化函数为“System.Text.Encoding.Unicode.GetString((byte[])dr ["字段名"])”，其中 dr 是使用的 OleDbDataReader 对象，而“字段名”即大二进制类型的字段名。

在文章显示页面使用 C#中的一个 Label 控件显示转化完成的 html 代码，则所保存的包括文本、图片、表格在内的全部内容即可在网页上展示出来。

2.2 文章的分章分节

分章节的目的，是为了浏览文章的时候能快速定位到所需章节。因此需要在所保存的文本中插入代表章节号的特定标签，以使用户在浏览时能够通过指定标签而使窗口自动移到该章节的起始位置。

本文采用的技术是使用网页标签“<h5></h5>”作为章节标签。添加标签的时候给它一个唯一的 ID，在打开该文章展示页面的连接后面加上“#”号和 ID，则浏览窗口会自动移动到该章节 ID 所在的位置。如加入完整章节标签“<H5 id=新章节>新章节</H5>”，即表示该标签 ID 为“新章节”，标签内容为“新章节”，在打开该文章展示页面的连接后面加上“#新章节”，则打开页面后，浏览窗口会自动移动到以“新章节”为首行的位置。图 3 为在编辑框中添加章节标签的效果，其中“<H5 id=第九章新添示例章节>第九章 新添示例章节</H5>”为新添的标签。图 4 为在浏览器中展示该文章并跳到指定章节的效果，地址栏中的地址后面的“#第九章新添示例章节”表明打开该文章后，窗口移动文章显示“<H5 id=第九章新添示例章节>”的位置。

添加的操作可通过 javascript 实现，在 FreeTextBox 控件编辑区中插入标签的 javascript 函数为：document.Form1.FreeTextBox1.document.selection.createRange().pasteHTML("标签")其中“FreeTextBox1”是页面中 FreeTextBox 控件的 ID，“标签”则是根据用户输入章节名所生成的准

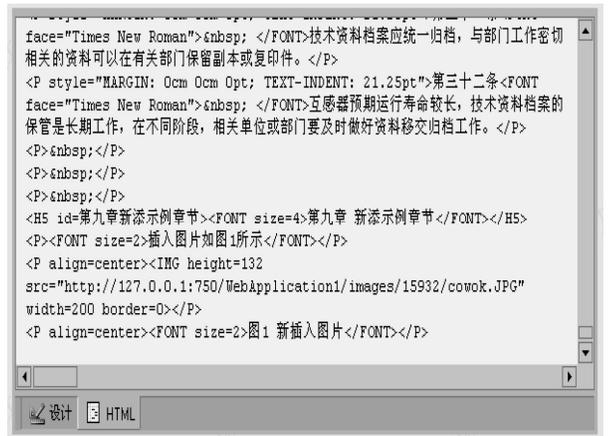


图 3 在编辑框添加章节标签



图 4 在浏览器中展示文章并移动指定章节

备插入 FreeTextBox 控件中的完整章节标签。在使用这个函数时，光标必须放在 FreeTextBox 控件编辑区中，否则会报错，因此需要手动捕捉异常，以提示用户放置光标。用户在一个文本框中输入章节名，然后在 FreeTextBox 控件的编辑区需要插入标签的位置放上光标，点击“添加”按钮触发 javascript 代码，用文本框里的内容作为标签 ID 和标签内容来生成一个完整的标签，并把它插入到光标的位置。

为方便用户选择文章章节，需要把文章的所有章节名罗列出来。为了更直观地表示章节间的层次关系，选用树形列表 treeview 来记录并显示章节名。文章的章节标签不能从大二进制类型字段中提取出来，因此须另外建表保存。在文章保存之前，把当前已添加的

章节名记录到 **treeview** 上；文章保存的时候再把 **treeview** 上的章节名及其层次关系保存到数据库对应的表中。**treeview** 上的节点和文章中的章节标签是一一对应的，因此添加、修改、删除都应该让 **treeview** 和 **FreeTextBox** 同步执行。实现方法如下：

①添加：在往 **FreeTextBox** 编辑区添加章节标签之前，用户需要在 **treeview** 上选定一个节点作为新添加的节点的父节点。在点击“添加”按钮时，先在 **FreeTextBox** 编辑框中添加章节标签，再于 **treeview** 上所选节点之下添加一个新章节节点，节点的内容为文本框中用户所输入的内容，和新章节的标签 ID 和标签内容相同。

②修改：用户在 **treeview** 上选定所要修改的节点，并在文本框内输入新节点的内容，然后点击“修改”按钮把选定节点的内容换成文本框的内容。同时，需要把 **FreeTextBox** 编辑区里原来的章节标签里的标签 ID 和标签内容换成文本框的内容。由于 **FreeTextBox** 具备文本编辑功能，原来的章节标签里可能夹杂了一些用户后来添加的文字样式标签，因此需要用正则表达式把被修改过的原章节标签找到，并完全替换成根据文本框内容新生成的完整章节标签。例如，C#正则表达式“<H5 id=原章节.*?</H5>”匹配以“<H5 id=原章节”开头，以“</H5>”结尾的，中间不再出现“</H5>”的一个字符串，即章节 ID 为“原章节”的完整标签。

③删除：用户在 **treeview** 上选定所要删除的节点，然后点击“删除”按钮把所选节点从其父节点的节点集删除。同时，需要把 **FreeTextBox** 编辑框里原来的章节标签删除。同样通过正则表达式找到原章节标签，用空字符串替换。

另外，在 **treeview** 中某个节点下添加子节点，只能添加到所有已存在子节点的后面，而不能随意选择插入的位置。为此设计两个按钮，用来向上和向下移动节点，改变它与兄弟节点的排列次序。用户选择好所要移动的节点，点击按钮使它与临近节点交换节点内容即可。这两个按钮只是改变 **treeview** 中的节点顺序，并不改变 **FreeTextBox** 编辑框里的章节标签顺序，因此 **FreeTextBox** 不执行任何操作。

用户保存文章的时候，也需把 **treeview** 记录的节点内容保存到数据库里的章节信息表中。可以设计一个递归函数实现 **treeview** 的深度优先遍历以保存信息。递归函数内容为：把当前节点的信息作为一条记录插入到数据表中，并对该节点的每一个子节点执行一次递归函数。而用户修改文章的时候，也需要在修改页面上的 **treeview** 中罗列所改文章现有的章节节点。同样可以设计一个递归函数深度优先遍历加载。递归函数的内容为：根据节点内容在 **treeview** 上其父节点下添加一个新的节点后，搜索数据库的章节信息表找到它的所有子节点的信息，并对每一个子节点信息执行一次递归函数。

2.3 树形列表浏览

为了方便用户浏览系统中所存储的文章，通过树形列表 **treeview** 展示所有文章及其章节。树形列表通过一个递归函数深度优先遍历加载。但是假如保存的文章数量较大，并且每篇文章下有多级章节，使用递归函数加载所有信息必定会频繁地连接数据库搜索节点信息，消耗大量的时间，因此，采用动态加载的办法来展示用户所需的文章及其章节。

所谓动态加载，是指在 **treeview** 初始化的时候并不加载并展开全部节点，而只是展示顶层的部分节点，当用户展开某个未曾展开的节点时，才搜索数据库获取其子节点信息，加载所有子节点。这样根据用户的需求来搜索数据库展示节点信息，大大减少了连接数据库的次数，节省了大量的时间。因为只有包含子节点的情况下，**treeview** 节点才能触发展开事件，所以加载的每一个新生成的子节点下均会带一个默认子节点。当某个节点触发打开事件时，如果该节点只有一个默认子节点，则证明它未被展开过，需要去掉默认子节点后，搜索数据库正式加载子节点。

TreeView 所展示的信息除了文章名和章节名外，还应包含一定的链接信息，以使用户通过点击 **treeview** 节点直接跳转到与该节点一一对应的文章或章节。在动态加载节点的时候，可以设置节点的 **NavigateUrl** 属性和 **Target** 属性，让节点变成一个超链接，点击该节点便可在 **Target** 指定的窗口打开 **NavigateUrl** 里的链接地址。链接地址的生成如 2.2

提到的,是由该文章展示页面的链接地址加上节点对
应章节的标签 ID 所组成。树形列表示意图见图 5。图
中左部为展示文章章节结构的树形列表,右部为展示
列表当前节点所对应文章内容的窗口,下部状态栏的
内容为列表当前节点的链接信息。

3 实际应用

通过此项技术,用户可以任意对文章进行分章分
节,使管理文章更加自由更加灵活。而且,不管原来
文章以什么格式保存在计算机上,用户均可以以“复
制”“粘贴”操作,把除图片以外的内容添加到编辑器
中自动转换成 html 代码并保存到数据库,而图片分
离出来后便可单独上传插入到相应的位置,这意味着

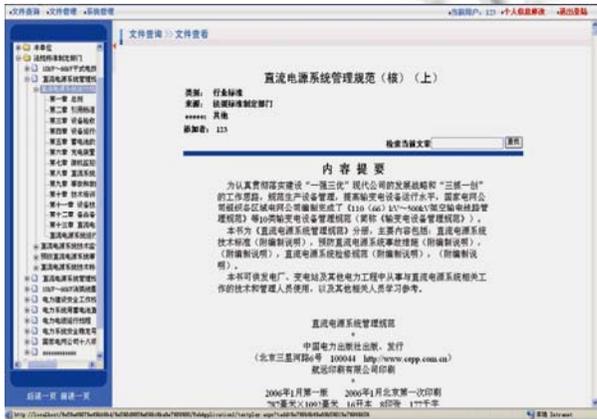


图 5 通过树形列表展示文章结构

任何本文格式的文章均不需进行复杂的格式转换,便
可用统一的格式保存到数据库中。

这种把大容量文章进行分章保存于数据库中并能
以树形列表展示其章节的技术,与原有的文章存储技
术比较起来,显得更方便,更安全。

4 结束语

本文提出的自定义分章大容量图文混编文章存储
技术是针对当前人们常用的一些文章管理系统的一些
不足之处而提出的改进办法,目的在于设计一种容易
实现,功能全面的文章管理技术,更方便人们的信息
存储和交流。该技术适用于一般的文章管理系统和电
子书刊。目前该技术已在中国电力出版社的《变电站
规程管理软件》中首次使用,得到了客户的认可,并
正式出版发行。

参考文献

- 1 崔俊杰.动态网页技术浅析.长沙民政职业技术学院学报,2006,13(1):104-106.
- 2 郑诗锋.基于WEB的文章管理系统的实现及探讨.武汉科技学院学报,2006,19(12):44-49.
- 3 廖金辉,李景福.Access 数据库中 OLE 对象的巧妙插入.电脑开发与应用,2004,(4).