

基于 XML 元数据交换的电子试卷管理系统

王海燕¹, 周思方², 支建飞¹

(1. 商丘师范学院计算机科学系, 商丘 476000; 2. 商丘职业技术学院计算机系, 商丘 476000)

摘要: 基于 XML 元数据交换算法, 在现有考试系统基础上, 增加将电子试卷输出为 XML 文档的功能, 设计出电子试卷管理系统。该系统按事先定义的 XML 文档模板, 从数据库中提取出学生的电子试卷, 生成包含学生原始答题信息和教师评卷信息的 XML 文档。实验结果表明, 该系统使电子试卷文档不再依赖现有考试系统, 加强电子试卷后期的管理。

关键词: 元数据; 交换算法; 电子试卷管理

E-paper Management System Based on XML Metadata Exchange

WANG Hai-yan¹, ZHOU Si-fang², ZHI Jian-fei¹

(1. Department of Computer Science, Shangqiu Normal University, Shangqiu 476000;

2. Department of Computer, Shangqiu Polytechnic, Shangqiu 476000)

【Abstract】 This paper designs an e-paper management system. It adds the function to bring output of e-paper into XML document based on the existing examination system. It is based on XML metadata exchange algorithm. The system extracts e-paper of student from the database to generate the original answer of students and XML documents of information marked by teachers according to the predefined XML document template. Experimental result proves that the e-paper documents can no longer rely on the existing examination system, and it enhances the later period management of e-paper.

【Key words】 metadata; exchange algorithm; e-paper management

1 概述

无纸化考试系统正逐步被很多高校接受与使用, 无纸化考试使学生的考试试卷数字化, 变成无形的东西, 而考试系统是基于数据库的, 试卷信息是数据表中若干条记录。传统的考试方式使用的纸质试卷是有形的, 学生的答题情况与教师的评卷情况一目了然, 有改动很容易检查, 试卷后期的管理也很直观、简单。而目前的无形化考试系统却忽视了该点。无形的数字化试卷管理存在以下问题: (1) 教师评分后试卷不再保存, 它很容易使考试流于形式, 教师为学生所评定的成绩也无据可查; (2) 保存考试系统的数据库, 这样看起来很完整, 但一旦现有考试系统升级换代或发生改变, 现有考试信息将不能再现; (3) 将电子试卷全部打印出来, 又违背了无纸化考试的初衷。因此, 对无纸化考试的后期试卷管理必须寻求一种合理、可行的方案。

本文结合基于 XML 文本格式元数据的交换算法^[1], 利用 XML 在文档描述方面的优势与特点, 对现有的考试系统进行改进。

2 基于 XML 元数据的交换算法

2.1 算法原理

基于 XML 元数据^[2]交换算法是使数据交换各方之间按共同规则描述元数据信息的 XML 模板文档, 通过解析 XML 模板文档识别该元数据的信息, 完成对元数据的存取交换功能, 而无须知道各应用程序的元数据库结构信息和元数据交换握手信号的规则等技术细节。图 1 描述了基于 XML 元数据交换算法的原理^[3]。元数据交换算法包括应用程序、XML 解析器组件和 XML 模板等部分。其中, XML 解析器组件实现对元数据的读写操作, 并向应用程序提供若干接口; 应用

程序可以利用组件提供的接口实现对元数据的写入和读取, 而无须知道读写的细节; XML 模板文档由应用程序的开发者提供, 向组件提供要读写变量的详细信息。XML 模板文档相当于应用程序和组件共同遵守的一个用来描述元数据信息的规则集, 组件可以通过解析 XML 模板文档来辨别出应用程序中需要交换的元数据信息。应用程序通过调用组件公开的接口来对元数据进行存取操作, 而无须知道实现存取的细节。组件根据接口参数传入的元数据名称, 在 XML 模板文档中找到相应的元数据信息, 然后识别出它的具体信息, 从而实现对该元数据的存取。由于组件通过 XML 模板文档推断出要交换存取的元数据信息, 因此组件端一旦交由应用程序使用, 就可以不加改变地交换不同应用程序中的元数据。

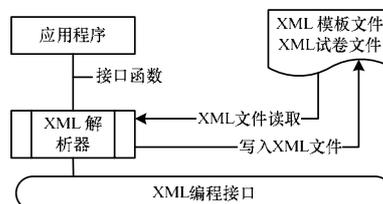


图 1 基于 XML 元数据交换算法原理

2.2 创建 XML 模板

XML 文档模板的创建原则是各应用程序和组件端之间

基金项目: 教育厅科技攻关计划基金资助项目“基于 Web2.0 网络课程开发平台”(2009A520021)

作者简介: 王海燕(1972-), 女, 讲师、硕士, 主研方向: XML 技术及网络教学; 周思方, 讲师、硕士; 支建飞, 讲师

收稿日期: 2008-12-09 E-mail: zxy98@sohu.com

必须遵循统一规则。电子试卷生成规则的 XML 文档模板中元数据信息采用类对象描述方式，统一表达为

```

<testDocument>
  <baseInfo>
    <courseName></courseName> //课程名称
    <student Name="" stuNo=""></student> //学生姓名与学号
  </baseInfo>
  <pageSetting>
    <paperSize height="" width=""></paperSize> //纸张大小
    <magin top="" bottom="" left="" right=""></magin>//页边距
    <layout></layout> //纸张方向
  </pageSetting>
  <examInfo>
    <items count="" score="" type="" lastscore="" num="">//题型、
//小题数量、分值、最终得分、题号
    <item score="" lastscore="" num="">//小题、分值与得分、题号
      <topic></topic> //题干
      <options count="">//选项，非选择题一般不用此节点
        <option></option>
      </options>
      <answer type="" doctype=""></answer> //学生答案
      <standardanswer type="" doctype=""></standardanswer>
//此题标准答案
      <description></description>
//教师评判，非客观题的评分情况说明
    </item>
  </items>
</examInfo>
</testDocument>
<hashCode>
<student></student> //学生交卷后生成的数字签名
<teacher></teacher> //教师评判后生成的数字签名
</hashCode>

```

其中 testDocument 是 XML 文本模板文件交换元数据部分的开始标签；baseInfo, pageSetting, examInfo 是 3 个基本类信息标签，baseInfo 存储课程名称与学生基本信息；pageSetting 存储电子试卷的页面设置信息用于控制试卷的显示与打印，examInfo 存储试卷的详细信息，包括试题信息，学生的作答情况和教师的评判情况；hashCode 标签用于保存考试试卷的数字签名信息，保证电子试卷的完整性，防止试卷被恶意篡改。在学生考试过程中生成试卷时，标准答案项与教师评判项标签均为空，在教师评卷时此类信息才会被加入到试卷中，以供打印或评估时查阅。

2.3 存取 XML 元数据的接口函数

各应用程序之间均采用统一规则的 XML 文本流模板作为元数据交换的接口标准，通过 XML 解析器对不同应用程序中的元数据进行解析与处理，此类操作与应用程序本身无关^[4]。应用程序与 XML 解析器之间通过接口函数实现数据存储。该接口函数的形式说明如下：

```

bool SetKeyValue(XmlDocument xmlDoc, string keyName, string attributeName, string stringValue)

```

其中，参数 xmlDoc 表示要操作的 XML 文档；keyName 表示要存储的节点名；attributeName 表示要设置的节点 keyName 中所包含的属性名；stringValue 为要存储的数据。

应用程序通过调用接口函数 SetKeyValue(xmlDoc, keyName, stringValue)向 XML 解析器传入参数，XML 解析器在 XML 模板文档中搜寻名称为 keyName 的元素或元素属性，

找到后再对该元素或其包含的属性赋值。SetKeyValue 的实现关键代码如下：

```

XmlNode xmlNode=xmlDoc.SelectSingleNode(keyName);
if (attributeName==string.Empty) {
  xmlNode.InnerText=stringValue;
}
else {
  xmlNode.Attributes[attributeName].Value=stringValue;
}

```

3 电子试卷的生成策略

3.1 生成电子试卷的 XML 文档

学生登录到考试系统后，系统会按约定的组卷算法为每个登录的学生生成试卷，组卷信息存储在数据库中。学生发出开始答题请求后，考试系统将学生的试卷信息从数据库中抽取出来，以数据集或内存表的序列化后的形式回发给客户端。出于对系统性能的考虑，在为每个学生组卷时并未立即生成对应的 XML 试卷，其原因是此时直接读写 XML 会消耗较多的服务器资源，当参加考试的学生较多时服务器资源的消耗更为明显。学生在交卷后，作答信息也暂时存储在服务器端数据库中，待一批考生考试结束后，由系统自动或管理员手工生成考生的电子试卷文档，此文档以 XML 的格式保存，其结构完全按模板结构输出，并为每一份试卷文档自动加入数字签名信息。

教师评判学生的试卷时，对客观题部分采用系统自动评分，对于主观题部分，一般采用人工评判。教师根据题目的标准答案给出得分，这与传统的评卷方式类似，但是对一部分比较复杂的主观题要求教师给出相对详细的评判情况说明。教师评卷结束后，系统自动计算出学生的总分，并将得分情况与评卷的情况说明一起保存到电子试卷的 XML 文档。

由于 XML 文档是纯文本格式，因此可以很容易被打开修改。为了保证试卷不被意外修改，将学生的答题情况与教师的评判情况如实地保存在试卷的文档中，在教师评卷结束后，对文档重新进行数字签名。

3.2 试卷文档生成过程的实现

试卷的生成过程由 2 个函数实现：(1)CreateDocumentAfterTest 函数；(2)SaveDescriptAfterJudge 函数。CreateDocumentAfterTest 函数算法流程如图 2 所示。

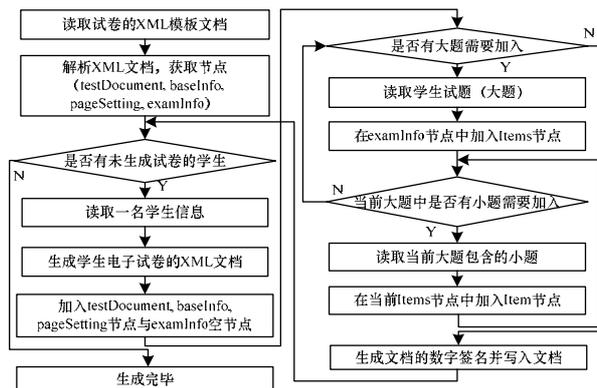


图 2 CreateDocumentAfterTest 函数算法流程

CreateDocumentAfterTest 函数的格式说明如下：

```

void CreateDocumentAfterTest(){...}

```

调用该函数将会为参加考试的所有学生生成对应科目的电子试卷文档。

SaveDescriptAfterJudge 函数算法流程如图 3 所示。

(下转第 276 页)