Design of Test Paper Automatic Generation Based on C#

Dong-Sheng ZHOU1,a,*, Fei PAN2,b, Jia-Ming GAO1,c and Mao-Ye XU1,d

1College of Information & Electrical Engineering, Shenyang Agricultural University, Shenyang, Liaoning, China
2Network Center, Shenyang Agricultural University, Shenyang, Liaoning, China

a zhoudsh8021@sina.com, b sie2128@163.com, c 616929139@qq.com, d 1075233711@qq.com

Keywords: Test Paper, C#, Word, Database, Duplicate Checking, Automatic.

Abstract. This paper mainly introduces a system about automatic test paper generation and duplicate checking based on the C# and SQL server 2008. The discuss how to use C# to operate Microsoft word 2010, including character formatting, domain controller, duplicate checking of word documents etc. Through the combination of C#, SQL Server 2008 and Word 2010, C# operates SQL server and read the data from it, outputs the data to the Word documents and formats the text. The software reduces labor capacity, the teacher only input the examination data and don’t consider the text format. Using the VBA programming, realizes the duplicate checking of over two Word documents, provides the repetitive rate. If the rate is over the setting rate, then the software can give the alert.

Introduction

The teachers give test paper about their subject at the end of the semester. Ordinarily, they need to provide the electronic test paper and the relative department prints it. The test paper has a fixed format, but teachers constantly change the format when they edit the test paper. In order to solve the problems, we develop the software and then teachers only need to input the type of test question and the demand of the test question, the system will generate the test paper automatically according to the test paper format. After generated test paper, the system can check automatically the similar degree of the two papers. If the similar degree is above the limited degree, the system will give an alert and then return the test paper input program and modify the test paper in order to reach the demand of similar degree.

System Analysis and Design

The system includes two departments, one is the generating test paper, the other is similar degree checking. The first part reads the data from the database and writes the data to the word document according to the format of test paper; the second part compares the two test papers, if the similar degree is blow the limited degree, then generates the ZIP format file, otherwise returns to the first part and generates the test paper again. The system structure is shown in Fig. 1.

The system adopts SQL Server 2008 as the DBMS. And the programming adopts C#. The database is the important part of the system, is the basis of the whole system. The followings are the structure of several important tables.

The table of individual_choice stores the test questions of individual choice; the table of Multiple_choice stores the test questions of multiple choice; and the other_question stores the other type of test questions, including calculation questions, short-answer questions and essay questions etc, the field of description is the type of the question, is used to generate the test paper. The table structures are shown in Table 1-3.
Key Technology Choice

The system adopts .NET, backend database is SQL Server 2008, programming language is C#, and operates the Office Word 2010, realizes to generate automatically the test papers of Word format.

Table 1. Table of individual_choice.

<table>
<thead>
<tr>
<th>Name of field</th>
<th>Description of field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Topic request</td>
</tr>
<tr>
<td>ChoiceA</td>
<td>A choice</td>
</tr>
<tr>
<td>ChoiceB</td>
<td>B choice</td>
</tr>
<tr>
<td>ChoiceC</td>
<td>C choice</td>
</tr>
<tr>
<td>ChoiceD</td>
<td>D choice</td>
</tr>
</tbody>
</table>

Table 2. Table of Multiple_choice.

<table>
<thead>
<tr>
<th>Name of field</th>
<th>Description of field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Topic request</td>
</tr>
<tr>
<td>ChoiceA</td>
<td>A choice</td>
</tr>
<tr>
<td>ChoiceB</td>
<td>B choice</td>
</tr>
<tr>
<td>ChoiceC</td>
<td>C choice</td>
</tr>
<tr>
<td>ChoiceD</td>
<td>D choice</td>
</tr>
</tbody>
</table>

Table 3. Table of Other_question.

<table>
<thead>
<tr>
<th>Name of field</th>
<th>Description of field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Topic request</td>
</tr>
<tr>
<td>Type_description</td>
<td>Describe the type of the test question</td>
</tr>
</tbody>
</table>

Based on the .NET framework module, C# is the development language, combined the character of C/C++ and Visual Basic, the function more great, the operation more simple. SQL Server 2008 affords the platform of the data management and analysis platform. Through ADO.NET data access structure, visits safely the database. The .NET platform system structure is shown in Fig. 1.

Figure 1. .NET Platform System Structure.
**Key Technology Application**

The program adopts the combination C# and VBA based on the SQL Server 2008. The key technology includes mainly database connection, document formatting and document comparing. Database connection strings is the basis of the program, the programming code is shown in the following:

```csharp
SqlConnectionStringBuilder scsb = new SqlConnectionStringBuilder();
scsb.IntegratedSecurity = true;
scsb.InitialCatalog = "test";
SqlConnection myConnection = new SqlConnection(scsb.ConnectionString);
```

Document formatting strings include MS Word 2010 checking and calling, the character formatting, page setup, adding domain control, document saving.

MS Word 2010 checking code is shown in the following:

```csharp
private static bool IsWordInstalledByVersion(string strVersion, RegistryKey machineKey)
{
    try
    {
        RegistryKey installKey = machineKey.OpenSubKey("Software").OpenSubKey("Microsoft").OpenSubKey("Office").OpenSubKey(strVersion).OpenSubKey("Word").OpenSubKey("InstallRoot");
        if (installKey == null)
        { return false; }
        return true;
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
        MessageBox.Show(e.StackTrace);
        return false;
    }
}
```

MS Word 2010 calling is very important, adds reference of Word, and defines variables of Word reference, the code is shown in the following:

```csharp
using Microsoft.Office.Core;
using Microsoft.Office.Interop;
using MSWord=Microsoft.Office.Interop.Word;
MSWord._Application wordApp; //Word application variable
MSWord._Document wordDoc;   //Word document variable
MSWord.Paragraph wordPara;   //Word paragraph variable
```

The character formatting programming code is shown in the following:

```csharp
shape.Application.Selection.Font.Bold = 1;
shape.Application.Selection.TypeText("Course name:__________________");
shape.Application.Selection.TypeParagraph();
```

The page setup finishes the page size, page margins and page alignment, the programming code is shown in the following:

```csharp
wordApp.ActiveDocument.PageSetup.TopMargin = wordApp.CentimetersToPoints(float.Parse("0.6"));
```
Domain control has many type, including textbox, art words and etc. Here takes textbox as a sample, the programming code is shown in the following:

```csharp
shape = wordDoc.Shapes.AddTextbox(MsoTextOrientation.msoTextOrientationHorizontal,
wordApp .CentimetersToPoints (float.Parse ("1.5")),
wordApp .CentimetersToPoints (float.Parse ("0.6")),
wordApp .CentimetersToPoints (float.Parse ("6.99")),
wordApp .CentimetersToPoints (float.Parse ("2.92")));
```

Word document saving Programming code is shown in the following:

```csharp
object format = MSWord.WdSaveFormat.wdFormatDocumentDefault;
wordDoc.SaveAs(ref path, ref format, ref Nothing, ref Nothing,
wordApp.Quit(ref Nothing, ref Nothing, ref Nothing);
```

Conclusion

Through coding the test paper automatic generation, we have understood more deeper C# and VBA, improved the ability of programming. At the same time, the program also raise the level of office automatition, and promoted the information level to a more higher level.

References


