Knowledge Assessment Software in Mining Specialist Training

Vladimir Lebedev[1], and *Olga* Puhova[1]

[1]Tver State Technical University, A. Nikitin Street 22, 170026, Tver, Russia

Abstract. The article reviews the knowledge assessment software module of electronic teaching and testing in mining specialist training. To develop the software module integrated programming environment state-of-the-art is used. Its advantage consists in small computer resource consumption, simple editing, and protection against the users' trying to find out the correct answers to test tasks. The software makes it possible to learn the ongoing learning information systematically and consistently as well as to assess the current knowledge in mining. The developed module meets the following requirements: a software module user-friendly interface, the storage of passed test results to be used for subsequent viewing, analyses, and evaluation, fast troubleshooting in case of any troubles with a stable module operation, and further software function extension and upgrading.

1 Introduction

To ensure the competitiveness and employability of young mining specialists of graduate open-cast education the quality of higher education should be improved. The modernization of Federal State Educational Standards calls for some software [1-3] meant for studying technological and industrial material independently [4-10] with the test of the knowledge gained to follow up and the human-generated interference to be eliminated.

When testing in a classical way an examiner is likely [11] to misunderstand the answer or not to take the correct answer in the account due to some possible nuances. As a result the test will not be passed in spite of correct answers. Software is free from such misunderstanding. Another important advantage of the software module is the possibility to calculate test results automatically and store all the information for subsequent analysis and qualitative assessment.

In line of rapid implementation and development of modern information technologies [12-14] the introduction and use of such software has many significant advantages. You can:

- study new materials prepared by a lecturer systematically.
- test and assess students' knowledge distantly.
- learn some information independently.
- reduce the time of current knowledge assessment considerably.

2 Materials and methods

To develop the software of students' knowledge assessment we used the state-of-the-art integrated programming environment such as Microsoft Visual Studio and EmEditor since they consume small computer resources and have reliable protection against trying to crack program source files. Visual Studio provides the development of various applications with a wide variety of functions and GUI functionalities. Text editor EmEditor allows different file formats to be edited and supports a wider range of functions than the standard means of Windows.

To implement the interface of electronic knowledge assessment software we choose the standard interface of Windows applications implemented with Windows Forms. The application advantages are as follows:

- software development ease and convenience
- typical controls
- small memory requirements which provides the opportunity to install the software on different computers.

For convenience the software module has several functional windows:

- a main window of students' training
- a window of users' authorization
- a window of students' knowledge testing
- a window of test result logging.

Such structure reduces the number of controls in each window and allows software functions to be distributed in order to ease and simplify the software learning by the users.

Each application window is made with individual Windows form.

The main controls used in developing the software were chosen as follows: Button, Label, Treeview, TextBox, RichTextBox, NumericUpDown, RadioButton, ListBox.

The authorization window allows a student to enter their personal data (a full name, a student group) so that the application module records the data in the result log after testing to maintain test result reports. The size of the window can be neither changed nor closed with standard ControlBox. The students are signed in only once when launching an application, with the data being kept as long as the program is active.

The window of students' testing allows the level of knowledge after completing the material study to be checked. Each time when finishing the test the program enters the result in the general table.

The testing window provides the user with the following functions:

- To choose one of the several tests proposed.
- To view the general test result table.
- To move to other application windows or to exit the program.

Being user-friendly the window displays the question number and the section title of the test being passed.

When the window of the electronic testing system is downloaded, the code executed initially is the following:

```
public Form2()
{
InitializeComponent();
label1.Visible = false;
button1.Enabled = false;
radioButton1.Visible = false;
radioButton2.Visible = false;
radioButton3.Visible = false;
radioButton4.Visible = false;
radioButton5.Visible = false;
button6.Enabled = false;
```

```
}
private void Form2_Load(object sender, EventArgs e)
{
NewFile = File.ReadAllLines(Application.StartupPath + «\\config\\2.dat»);
}
```

The code prevents visibility or access to some controls unused at the moment. Function Form_Load downloads the data of folders storing the question and answer option files into array NewFile from the file. This allows new tests to be added and available tests to be edited.

The array is a data structure of some number of variables accessed by indices computed. The array variables are its elements and have the same type called a type of array elements. In programming the numeration usually starts with zero while the test numbers start with one. Thus the users were restricted to select the test number not less than one and not more than the total test number is.

The code uploads the titles of main training sections represented in a tree-like structure from the main application window table of contents (Fig. 1). Test titles are on the main (zero) level of a tree structure.

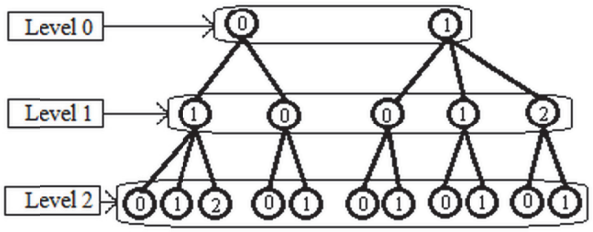

Fig. 1. Tree structure levels

The code downloads the total list of questions and displays one of them. A random sequence of questions is generated with random numbers.

For the user's convenience the window displays a question number and a section title being tested. The test number is selected with left corner buttons. Button Next Question at the bottom of the testing window uploads the next question from the test. загружает следующий вопрос из теста. At the end the program displays the appropriate message and writes the data obtained.

A result logging window displays the total list of all the tests ever completed.

The window is intended for a lecturer conducting testing and shows the test results with a comprehensive set of data related to each one being tested. The window is of a static size and has only one close button.

To log test results class Filestream is used. It represents Stream in a file and supports synchronous and asynchronous read and write operations, file opening and closing operations in a file system, and the change of other operating system descriptors when processing files. To achieve the best performance FileStream buffers I/O. It is important to remember that after the work with them the data streams should be closed to free the memory from their children. This is particularly important given the work with several children. The streams are closed with the following code:

```
srFromFile.Close();
fsFile.Close();
```

When the form is loaded, the code executed is the following:

```
string strText = ««;
FileStream fsFile = new FileStream(Application.StartupPath + «\\config\\res.dat», FileMode.OpenOrCreate, FileAccess.Read);
```

```
StreamReader srFromFile = new StreamReader(fsFile);
while (!srFromFile.EndOfStream)
{
strText = srFromFile.ReadLine();
listBox1.Items.Add(strText);
}
srFromFile.Close();
fsFile.Close();
```

The data read from the file are displayed string by string with element ListBox. The information is written in one string so there is no necessity to organize a big database to write and store test results.

3 Results and discussion

The test includes the tasks of various types, e.g. multiple choice, an answer keyboard input, matching, ordering, classifying. In online testing a tester sees the details on every tested student's progress on their computer. After the test the data are archived so that test results can be viewed and analyzed with the embedded application means. The application provides for test generation as offline executable files which can be distributed for testing offline and without storing the results. The mode is intended, first of all, for self-examination. To begin testing, simply start the output file on any computer with Windows OS, the installation of any applications is not required.

When a test type is selected and an answer for the first question is given, a user should click the button to choose the next question. The code being executed is the following:

```
RadioButton[] RB = { radioButton1, radioButton2, radioButton3, radioButton4, radioButton5 };
foreach (RadioButton rb in RB)
{
if (rb.Checked && rb.Text == right)
{
res++;
}
}
number++;
if (number < ansnum)
{
count++;
label3.Text = «Question: « + Convert.ToString(count) + « from « + ansnum;
if (count == ansnum)
{
button1.Text = «End Test»;
}
RND = new Random();
int k = RND.Next(0, L.Count);
Find(L[k]);
L.Remove(L[k]);
}
else
{
MessageBox.Show(«No more questions»);
number = 0;
```

```
double allans = ansnum;
double endres = res / allans * 100;
endres = Math.Round(endres, 2);
string test;
string verdikt;
if (endres >= 80)
{
test = «You passed the test»;
verdikt = « (The test is done)»;
}
else
{
test = «You did not pass the test»;
verdikt = « (The test is not done)»;
}
}
```

First a programmer checks if RadioButton has been chosen. RadioButton corresponds to the correct answer and increases the value of the variable meaning the quantity of correct answers. Then a new question is generated with function Find. When all the questions are answered, the application calculates the user's result and displays through MessageBox a message about the percentage and informs if the threshold of 80 % is passed which means that the test is passed.

When completed the application also writes the full data on the test passed into a file which can be viewed in the result table window. The application executes the following code:

```
resultate = «Student: « +Form3.fam + « « + Form3.name + « « + «Grope: « +Form3.group
+ « «;
resultate = resultate + «Test Number: « + Convert.ToString(numericUpDown1.Value) + «
«;
resultate = resultate + «Result: «+ Convert.ToString(endres) + «%» + verdikt +» Date: « +
DateTime.Now;
FileStream swFile = new FileStream(Application.StartupPath+»\\config\\res.dat»,
FileMode.Append, FileAccess.Write);
StreamWriter sw = new StreamWriter(swFile);
sw.WriteLine(resultate);
sw.Close();
```

The information obtained can be viewed after finishing the test by clicking Test Result Table.

4 Conclusions

The developed software module has a simple, convenient, and plain user interface with distributed functionalities. The software provides the possibility of fast editing and has the following functions:

- To show text and graphical information to study systematically.
- To authorize software users.
- To assess students' current knowledge through automated testing.
- To eliminate the human-generated interference when the test is being passed.
- To store test results through logging for subsequent viewing and analyzing.

All the tests and test results are coded which excludes the possibility of fraud and falsification completely. There is also a possibility of determining the ceiling of one test

passing number which raises the students' testing standards. The software module does not require large memory which allows it to be installed on any modern computers. There is also a possibility of quick troubleshooting and resolving any issue.

The software module of mining students' knowledge assessment (graduate education program 21.05.04 Mining, Open-cast specialization) has been successfully piloted at the Chair of Geotechnology and Peat Production, Tver State Technical University. The software of electronic testing system has received the Certificate of State Registration.

References

1. N. Nagashree, N.V. Pujari, Comp. Hum. Behav., **65**, 1-131 (2016)
2. E. Sarmiento Leite, E. Almentero, Sotomayor Alzamora, El. Not. Theor. Comp. Sci, **329**, 123-148 (2016)
3. T. Wang, Z. Zhang, X. Jing, L. Zhang, Autom. Soft. Engin., **23:4**, 569-590 (2016)
4. A.E. Afanas′iev, S.N. Gamayunov O.S. Misnikov, Colloid J., **61:3,** 274-279 (1999)
5. O.S. Misnikov, O.V. Dmitriev, V.I. Popov, E.Yu. Chertkova, Polym. Sci. D, **9:1**, 133-139 (2016)
6. O.S. Misnikov, O.V. Dmitriev, E.Yu. Chertkova, Euras. Min. Gornyi Zhurnal, **2:24**, 30-34, (2015)
7. O. Misnikov, Polym. Sci. D, **7:3**, 252-259 (2014)
8. O.S. Misnikov, E.Yu. Chertkova, Euras. Min. Gornyi Zhurnal **1:21,** 63-68 (2014)
9. O. Misnikov, A. Timofeev, O. Pukhova, Polym. Sci. D, **8:1,** 66-74 (2015)
10. O. Misnikov, Mires and Peat, **18:22**, 1-15, (2016)
11. K. Eck, G.P. Alleman, V. Quick, J. Martin-Biggers, N. Hongu, C. Byrd-Bredbenner, J. Com. Health **41:6**, 1187-11951 (2016)
12. F.N. Abu-Abed, D.V. Martynov, A.V. Ivanova, R.V. Dopira, R.Y. Kordyukov ARPN J. Eng. App. Sci., **11:16**, 9636-9645 (2016)
13. B.V Palyukh., V.K. Ivanov, I.A. Egereva, IJAER, **10:24**, 45736-45740 (2015)
14. Y.N. Matveev, B.V. Palyukh, N.A. Stukalova, V.N. Bogatikov, IJAER, **10:24**, 45717-45723 (2015)