2019 IEEE 4th International Conference on Computer and Communication Systems

Design and Implementation of Software Data Management on Acquiring Training System

Xuedong Xue[1], Xude Cheng[1], Wei Peng[1], Fei Lu[2] , Pin Wang[1]
[1]Ordnance NCO Academy, Army Engineering University, Wuhan Hubei, China
[2]Military Representative Office of the 207 Institute, Taiyuan Shanxi, China
e-mail: [1]xxdmymail1228@126.com

***Abstract*—A missile acquiring training system was designed and developed for the problems of large wear loss, no evaluation function, and limited site constraints in the live-fire training. Because of the data management module is a place for data storage of the training system, and the system uses a SQL Server database with high data confidentiality and database maintenance function to complete the storage and management of a missile acquiring training data. Therefore, this paper focuses on the methods of data viewing, data deletion, data export, data storage, data backup and data recovery, and solves the problems of openness, security, reliability, parallelism and operability of data management. Further, it provides an important reference for the development of other similar training systems.**

Keywords-aiming training simulation system; self-test calculation method; data output format conversion and precision control method

I. INTRODUCTION

A missile acquiring training system mainly simulates the function of the acquiring subsystem, and performs the acquiring training before the missile launch. The system consists of two parts: hardware and software. The system hardware is mainly composed of three parts: acquiring trainer, acquiring equipment and acquiring special computer. The system software mainly completes the functions of acquiring calculation and fault inquiry. The system software consists of three modules: acquiring calculation, data management and printing. The data management module is a place for data storage of the training system. The various operations of the system are database-centric. The database management and daily maintenance ensure the important operation of the system and data security. Data management provides data. View, data deletion, data export, data saving, data backup and data recovery[1,2].

The security and confidentiality of the database is a global problem. It is related to all aspects of the system. Due to the characteristics of the targeting training system, the security of the database includes identity authentication, access control and data backup[3,4]. The database of this system uses SQL Server database platform, which on one hand mainly considers the security mechanism of SQL Server database, such as data confidentiality, data integrity and transaction processing.[5] On the other hand, SQL Server comes with a powerful database maintenance function to avoid loss in the event of database security problems. You can restore the backed up data to the system[6,7,8].

II. DATABASE DATA MANAGEMENT

A. Data View

Call the database functions MoveFirst, MoveNext, MovePrev, and MoveLast to view the data. After each call to the above function, you must call the DisPlay function to display the data in memory. The fllowing is the code for DisPlay:

```
time_Text.Text = rs.Fields("m_data")
    a1_Text.Text = rs.Fields("m_a1")
    a2_Text.Text = rs.Fields("m_a2")
    a3_Text.Text = rs.Fields("m_a3")
    ……
    p_Text.Text = rs.Fields("m_p") 'Self-checking result
    index_Label.Caption = "Current" & CStr(m_rec_index) & "Record"
    On Error GoTo 0
```

B. Data Deletion

First call the Delete function and the update function to delete the current record, and then move the database pointer to the first record.

C. Data Export

Import the records in the current database directly into the calculation window in each Edit control。

III. FILE DATA MANAGEMENT

First read the file name of all text files in the specified folder to the list box, and then read the corresponding file according to the viewing requirements. The program settings are as follows:

```
If FileSystem1.Dir(strTemp, fsAttrDirectory) <> "" Then ' Check if the folder exists
  strFileName = FileSystem1.Dir(strTemp & "\*.txt", fsAttrNormal) ' Read the file name from
  Do While strFileName <> ""
   fileList.AddItem strFileName
   strFileName = FileSystem1.Dir
  Loop
 Else
  MsgBox " No data in the system ! ", vbCritical, "Acquiring calculation "
```

```
  End If
Read text file：
File1.Open FileName, fsModeInput, fsAccessRead
  Do While File1.EOF = False
   'strTemp = File1.LineInputString   ' After executing LineInputString, the file pointer will automatically advance
    File1.Seek = File1.Seek + 13 'The role of Seek is to jump forward a few pieces of data to eliminate the preceding spaces.
    strfileText = strfileText + File1.LineInputString + Chr(13) + Chr(10)
  Loop
  File1.Close
  fileText.Text = strfileText
```

IV. Data Saving

The system saves the calculated data in two ways: database and text file. The two methods are backups to ensure the reliability of data storage.

A. *Database Mode*

First check if the database file exists, if it does not exist, and create a new database file; secondly open the database, and create a new database record. Write the calculation data; finally close the database, and disconnect the database. The program settings are as follows:

```
 Set conn = Nothing
   IniDatabase ' Initialize the database
   If connOpen = True Then      The role of 'connOpen = True is to establish a database connection，
IniDatabase  Just check if the database file exists, if it doesn't exist, it will be established, and if it exists, then it will do nothing.
   strTime = Year(Date) & "年" & Month(Date) & "月" & Day(Date) & "日" & Hour(Time) & "时" & Minute(Time) & "分" & Second(Time) & "秒"
   Set rs = CreateObject("ADOCE.Recordset.3.0")
   rs.Open "miaozhunDataTable", conn, adOpenKeyset, LockPessimistic
   If Err.Number > 3000 Then' Open failure handling
    MsgBox " Failed to open table entry in database, can't save data！ " & " error code：" & Err.Number & " Content" & Err.Description, vbCritical, " Acquiring calculation "
    Exit Sub
   End If
   rs.AddNew
   rs.Fields("m_data") = strTime   ' The following is in the order of the fields in the database
   rs.Fields("m_a1") = a1_Text.Text
   rs.Fields("m_a2") = a2_Text.Text
   rs.Fields("m_a3") = a3_Text.Text
   .....
   rs.Update
   If conn.Errors.Count = 0 Then
    MsgBox " Save to database successfully ！ ", vbExclamation, "Acquiring calculation"
   Else
    MsgBox " Saving to database failed！ ", vbCritical, "Acquiring calculation "
   End If
```

B. *Text File Mode*

Build a new text file, generate a file name based on the current date and time, write the calculated data to it, and close the file.

V. Data Backup and Data Recovery

In the systems of storing a large amount of important data, it is especially important to back up the data in order to avoid unnecessary loss due to accidental data loss. For encapsulating the operation of the database through the application program, the database backup and recovery functions are integrated into the system, which greatly facilitates the operator and also enhances the stability of the system data.

The core code for data backup is as follows:

```
private void button1_Click(object sender, EventArgs e)
    {
       string Str_dar = "";
       Str_dar = textBox2.Text + "\\";
       if (textBox2.Text == "")
       {
  MessageBox.Show ("Please choose the backup path of the backup database file!");
       }
       try
       {
          Str_dar = "backup database qchs to disk='" + Str_dar + (System.DateTime.Now.ToShortDateString()).ToString() + MyMC.Time_Format(System.DateTime.Now.ToString()) + "_ Backup data.bak" + "'";
MyDataClass.getsqlcom(Str_dar);
          MessageBox.Show("Data backup is successful! ", "hint", MessageBoxButtons.OK, MessageBoxIcon.Information);
          this.Close();
       }
       catch (Exception ex)     {
 MessageBox.Show(ex.Message, " hint ", MessageBoxButtons.OK, MessageBoxIcon.Information);
       }
    }
```

The core code for data recovery is as follows:

```
 private void button5_Click(object sender, EventArgs e)
    {
        if (textBox3.Text == "")
       {
MessageBox.Show("Please select the database file to be restored！ ");
          return;
       }
```

```
        try
        {
          if (linkdata.My_con.State == ConnectionState.Open)
          { linkdata.My_con.Close();
          }
          string DateStr = "server=127.0.0.1,1433\\SBCServer;database=master;uid=sa;pwd=sa";
          SqlConnection conn = new SqlConnection(DateStr);
          conn.Open();
          //------------------- Kill all processes that connect to the qchs database ----------
          string strSQL = "select spid from master..sysprocesses where dbid=db_id( 'qchs') ";
          SqlDataAdapter Da = new SqlDataAdapter(strSQL, conn);
          DataTable spidTable = new DataTable();
          Da.Fill(spidTable);
          SqlCommand Cmd = new SqlCommand();
Cmd.CommandType = CommandType.Text;
          Cmd.Connection = conn;
          for (int iRow = 0; iRow <= spidTable.Rows.Count - 1; iRow++)
          {
Cmd.CommandText = "kill " + spidTable.Rows[iRow][0].ToString();  // Force closed user process
            string a = spidTable.Rows[iRow][0].ToString();
            Cmd.ExecuteNonQuery();
          }
          conn.Close();
          conn.Dispose(); / SqlConnection Tem_con = new SqlConnection(DateStr);
          Tem_con.Open();
          SqlCommand SQLcom = new SqlCommand("backup log qchs to disk='" + textBox3.Text.Trim() + "' restore database qchs from disk='" + textBox3.Text.Trim() + "'", Tem_con);
SQLcom.ExecuteNonQuery();
          SQLcom.Dispose();
          Tem_con.Close();
          Tem_con.Dispose();
MessageBox.Show("Data restore succeeded!", "Hint ", MessageBoxButtons.OK, MessageBoxIcon.Information);
MyDataClass.con_open();
MyDataClass.con_close();
MessageBox.Show(" In order to avoid data loss, the entire system will be shut down after the database is restored! ", "hint", MessageBoxButtons.OK, MessageBoxIcon.Information);
          Application.Exit();
        }
        catch (Exception ex)
        {
MessageBox.Show(ex.Message, " Hint ", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
      }
```

VI. Implementation of Printing Function

Since the handheld computer does not have a standard printer interface, its USB port can only be used to communicate with a normal computer, so it is very difficult to print the calculation results using the usual method. After market research and many trials, the research team realized the printing function through two special methods.

A. Wireless Printing Method

You can choose a printer that supports infrared or Bluetooth. Where, the Bluetooth interface is more reliable, and the Bluetooth interface is a new short-range wireless communication technology. Two devices with Bluetooth interfaces can communicate in any direction within a radius of about 10 meters.

B. SD Card Transfer Printing Method

The SD card is similar to the popular storage devices such as USB flash drives and memory sticks, and is manufactured by CMOS technology.

The method: the first step is to convert the calculation result from text format to image format and save it to the SD card; the second step is to remove the SD card from the handheld computer and insert it into the card slot of the printer to start the printer to print the picture.

The algorithm for converting text format data to image format is as follows:

```
hdc=::CreateDC(str,NULL,NULL,NULL);  //"display"
bits=::GetDeviceCaps(hdc,BITSPIXEL)*(::GetDeviceCaps(hdc,PLANES));
  ::DeleteDC(hdc);
  ::GetObject(hbitmap,sizeof(BITMAP),(LPSTR)&bitmap);
bi.biSize=sizeof(BITMAPINFOHEADER);
  bi.biWidth=bitmap.bmWidth;
  bi.biHeight=bitmap.bmHeight;
  bi.biPlanes=1;
  bi.biBitCount=wbitcount;
  bi.biCompression=BI_RGB;
  bi.biSizeImage=0;
  bi.biXPelsPerMeter=0;
  bi.biYPelsPerMeter=0;
  bi.biClrUsed=0;
  bi.biClrImportant=0;
dwbmbitssize=((bitmap.bmWidth*wbitcount+31)/32)*4*bitmap.bmHeight;
hdib=::GlobalAlloc(GHND,dwbmbitssize+dwpalettesize+sizeof(BITMAPINFOHEADER));
lpbi=(LPBITMAPINFOHEADER)GlobalLock(hdib);
  //*lpbi=bi;
   if(lpbi==NULL)
     {
       CString strTemp=" Failed to allocate memory！ ";
```

```
        LPCTSTR lpstr=strTemp;
AfxMessageBox(lpstr,MB_OK,0);
     }
hpal=GetStockObject(DEFAULT_PALETTE);
  if(hpal)
  {
       hdc=::GetDC(NULL);
holdpal=::SelectPalette(hdc,(HPALETTE)hpal,false);
       ::RealizePalette(hdc);
  }
GetDIBits(hdc,hbitmap,0,(UINT)bitmap.bmHeight,(LPSTR
)lpbi+sizeof(BITMAPINFOHEADER);
  LPVOID  mBits=bitmap.bmBits;
  m_edit=sizeof(mBits);
  m_con_edit.UpdateData(false);
  m_con_edit.UpdateWindow();
  if(holdpal)
  {
       ::SelectPalette(hdc,(HPALETTE)holdpal,true);
    ::RealizePalette(hdc);
       ::ReleaseDC(NULL,hdc);
  }
  CFile file;
if(!file.Open(lpfilename,CFile::modeCreate
CFile::modeWrite,NULL))
  {
       CString strTemp=" Failed to specify directory
creation file！ ";
     LPCTSTR lpstr=strTemp;
 AfxMessageBox(lpstr,MB_OK,0);
  }
  bmfhdr.bfType=0x4d42;//"bm"
dwdibsize=sizeof(BITMAPFILEHEADER)+sizeof(BITMA
PINFOHEADER)+dwpalettesize+dwbmbitssize;
  bmfhdr.bfSize=dwdibsize;
  bmfhdr.bfReserved1=0;
  bmfhdr.bfReserved2=0;
bmfhdr.bfOffBits=(DWORD)sizeof(BITMAPFILEHEADE
R)+(DWORD)sizeof(BITMAPINFOHEADER)
file.Write( &bmfhdr,sizeof(BITMAPFILEHEADER));
file.Write( &bi,sizeof(BITMAPINFOHEADER));
file.Write(                &mBits,dwbmbitssize);//dwdibsize
dwbmbitssize
  GlobalUnlock(hdib);
  GlobalFree(hdib);
  file.Close();
}
```


ACKNOWLEDGMENT

After the successful development of a missile acquiring training system, it solved the problem that the military and colleges acquiring professional teaching and training didn't have supporting maintenance training equipment, and relied on the actual equipment for training, which had large loss of main equipment and low safety factor. At the same time, the shortcomings and vulnerabilities in the currently installed acquiring system and the reliability and maintainability of equipment use had been improved, which can meet the needs of military training and college teaching. Database is the core of information system, so the choice of database is a very important aspect of system development. Generally speaking, the factors that need to be considered in the choice of database platform are openness, security, reliability, parallelism and operability. This paper had given a data mining module implementation method for a missile targeting training system, which solved the problem of data management security and reliability, and provided an important reference for the development of similar training systems.

REFERENCES

[1] Xu Jianru. General Automatic Test System Design and Technical Research for Circuit Board [D]. Northwestern Polytechnical University, 2001.

[2] ZHANG Guo-yu, et al.An Optoelectronic Inspection Method for the Taper of a Bore[J].ACTA ARMAMENTARII,2007-2,28(2):108-109.

[3] Fang Zejun, Experimental Study of the Impact of Fuel Additive on Combustion & Emission Performance of Gasoline Engine [D], Beijing University of Technology, 2008.

[4] Dai Jianhua & Li Kaicheng, Measurement of High Current Based on Rogowski Coil [J], High Voltage Engineering,2002,28（1）: 1-4.

[5] LIANG Gang , WANG Chong-hua.SyStem modeling of port container crane training Simulator based on object-orientedmethod[J].CHINESE JOURNAL OF CONSTRUCTION MACHINERY,2006-1,4(1):108-109.

[6] Mohamad Sawan, Faycal Mounaim, Guillaume Lesbros. Wireless monitoring of electrode-tissues interfaces for long term characterization [J] .Analog Integrated Circuits and Signal Processing. 2008, 55(1).

[7] Wei Baohua, Yang Suochang , Zheng Silong et al, Development Model of Open Circuit Testing System , J. Ordnance Industry Automation, 2004,23(5):76-7.

[8] Liu Peng, PCB Online Testing Technology [J], China Instrumentation, 2003, (2):44-46.