

KLIJENTSKA FORMA ZA VIZUELNU KONTROLU PODATAKA IZ DVA IZVORA CLIENT FORM FOR VISUAL DATA COMPARISON FROM TWO SOURCES

Siniša S. Ilić, Stevan Radoičić *Faculty of technical sciences in Kosovska Mitrovica, Serbia*

Sadržaj - Prilikom razmene podataka, postoji slučaj da strana koja očekuje podatke prvo šalje fajl klijentu u dogovorenom formatu popunjen inicijalnim vrednostima. Po prijemu takvog fajla klijent po potrebi menja inicijalne vrednosti svojim podacima u istom fajlu i vraća ih strani koja očekuje podatke. U postupku promene podataka potrebno je izvršiti poređenje novih vrednosti sa inicijalno popunjenim. U ovom radu je predstavljena forma, napravljena u MS Access-u, kojom se vizuelno upoređuju navedene vrednosti. Forma za upoređivanje je podeljena na dva dela u obliku tablica u kojima se ćelije tablica pomeraju sinhronizovano. Ukoliko su vrednosti na istim pozicijama u tabelama različite, odgovarajuća polja menjaju svoju boju. Nakon izvršene provere, korisnik može da snimi nove vrednosti u bazu podataka i istovremeno kreira fajl u dogovorenom formatu sa novim vrednostima i pošalje ih prema strani koja očekuje te podatke.

Abstract - In the process of data exchange between two parties, there is a case when one party sends file in predefined format to other party with initial values. After receiving of the file, other party changes the appropriate initial values with correct values in the same file and returns back to the sender. There is a need that party that modifies data compare initial values with correct values visually before sending back. In this paper is presented the MS Access form that enables visual comparison of before mentioned values. The form is divided into two parts as two tables where navigating through the cells is synchronized. If the values of the cells in same position are different, the colors of appropriate fields are changed. After the control of the values is performed, the user can save the new values in the database, modify file in predefined format and send back to the sender.

Keywords – Data comparison, MS Access form, synchronization of sub-forms

1. INTRODUCTION

In multi-user information systems (IS), centralized data processing often depends partially or completely on batch loading input. In this case clients have to prepare needed data and to pack it in predefined electronic format. It is usually XML, CSV, ASCII or other formats.

The batch loading input must provide two types of validation on server side: format validation and data validation. Format validation verifies if data interface files sent from clients respects predefined format, i.e. if XML file is according to the XSD schema, if there is fixed number of characters per row in ASCII file in each row, if appropriate part of data in file is of expected data type etc. Data validation verifies the values of extracted data from interface file against the constraints according to the business rules of IS.

The first validation is usually performed by server side application and the second one usually compares loaded data with already existing data in database [1]. In any case, if data format or data itself is wrong, the centralized system needs to reject interface file. The client then, has to modify the format and/or content of the file and resend back it again. When there are more than 1000 clients that have to send interface files, the problem arises if all of them are capable to create interface files in predefined format. Also, the question

appears: if server rejects files are they capable to correct errors there quickly without professional help.

2. DEFINITION OF THE PROBLEM

The management of the centralized IS has decided to use formatted Excel files as interface files already filled with initial values. There are two advantages from this decision:

- It is easier for clients to work in Excel than to work in XML editors or to manually edit CSV or ASCII files,
- It is easier for clients to modify initially filled data (with default values, where in many cases default values can be correct values) compared to complete insertion of data.

The Excel file, sent to the client, is formatted in the way that some of the fields, that are typical for the specified client (client name, id and similar) are locked and cannot be accessed, and some other fields are filled with data assumed to be correct for the client. Usually clients get periodically excel files, modify the data there and upload modified files back to IS.

Problem that rises there is that clients have to prepare and keep data in separate Excel files. They cannot compare content of the files and build some statistics from them. Also, some of the source data is not entered in excel, but calculated by using last updated coefficients. Sometimes can happen that clients by mistake send wrong data because not the last

tables. The example of VBA source code that reads values of „E“ column cells within Do While loop until the cell is empty, and puts the values in database table is shown at figure Fig 3.

```

If oSheet.Cells(r, "E") > 0 And oSheet.Cells(r, "E") <> "" Then
    myRec.AddNew
    myRec.Fields("Entry") = oSheet.Cells(r, "E")
    myRec.Update
End If

```

Fig 3. Example of the VBA source code that reads the Excel cell value and puts the values in database table

In order to clarify the variables in source code presented, r is row number, E is column notation of Excel table and myRec represents the table row. The command myRec.AddNew adds a new row in database table and myRec.Fields("Entry") = oSheet.Cells(r, "E") assigns the value of the cell in row r and column E to the field Entry in the new record. The formats of Excel sheets are predefined so for each column in Excel there exists appropriate field in Access database, and for each sheet in Excel there exists appropriate table in Access.

In manual preparation of data in Excel to be sent to centralized database, clients are able to directly change the values in initially filled cells, if the correct value is not equal to initial value, and to leave initial values if it is. In some way it represents control of entered data. For each cell client compares initial values with prepared values to be entered. If client notice in some cell some logical deviation between initial and prepared for entry value he checks if his values are correct.

When a procedure of manual modifying of Excel sheet is over, there is a way to compare initial and modified values once again – Excel enables comparison of two excel files side by side, but client has to look carefully and compare the values in the same cells in different files. One file is the file sent from central database with initial values and another is the file with modified values. Moving through the cells in one Excel file is not fully synchronized with moving in another file. Only preview boundaries are synchronized i.e. if one can see contents of columns A-P and rows 10-17 in one file, the contest of the same set of columns and rows will be shown in another file. But there is not synchronized selection of the cell.

In our software application, after the Excel data is imported, the client form for visual data comparison is opened. The design of the form is similar to side by side [8] Excel comparison of the files, with improved synchronization of the cell selection and with visually marked differences between values in the cells in same positions from two sources. It is not only the preview of the differences, the user can also change the value in any cell, save changes in database and export to Excel file ready to be sent to centralized database system.

The form consists of two sub-forms in datasheet format that are placed one below the other. The navigation through the cells in both sub-forms is synchronized. Imported data from Excel sheet is shown in upper sub-form and data prepared for export is shown in lower sub-form. The design

of client form for visual comparison of data from two sources is presented in Fig. 4.

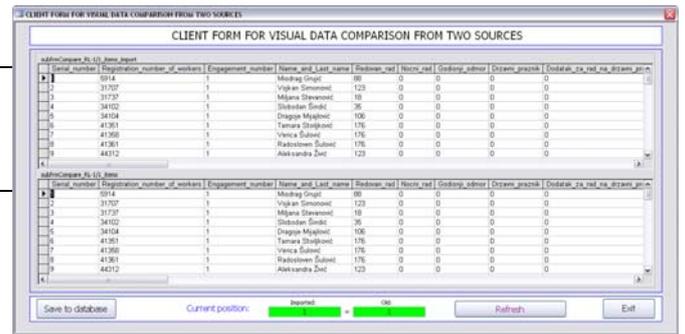


Fig 4. Client form for visual data comparison from two sources

We have used the conditional formatting to mark cells in upper sub-form for which the values are not equal to the values in cells in lower sub-form at the same positions [9]. The red background color is selected to mark these cells. The VBA source code that performs dynamic conditional formatting is shown at Fig 5.

```

Me!Serial_number1.FormatConditions.Delete
Set objFormCon = Me!Serial_number1.FormatConditions.Add(acExpression, ,
"iif(isnull([Serial_number1]),'xxx',[Serial_number1]) <>
iif(isnull([Serial_number]),'xxx',[Serial_number]))"
With Me!Serial_number1.FormatConditions(0)
    .BackColor = lngRed
End With

```

Fig 5. VBA source code that performs dynamic conditional formatting

In the process of visual comparison data from two sources client cannot modify values from upper sub-form, but he can modify only values from lower sub-form. When the value in some cell is modified, the background of the appropriate cell in upper sub-form is automatically changed. In the VBA source code, object objFromCon is defined that assigns the rule for conditional formatting through its property FormatConditions. When the rule condition is fulfilled the .BackColor property formats the cells. The look of the client form with enabled conditional formatting is shown at Fig. 6.

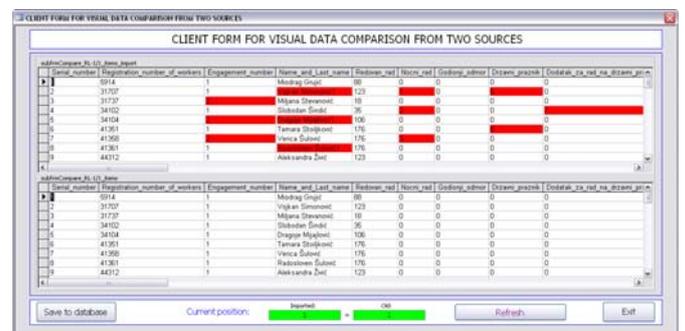


Fig 6. Automatical formatting of cell's background

An extra visual control is placed at the bottom of form. There are two text fields filled with the values of synchronized cells. The left field value is the value of the selected cell in upper sub-form and the right field value is the value of the cell at the same position in lower sub-form. If the values of the left and right text field are equal, both fields

have green background. But, if the values are not equal backgrounds become red.

As it is already mentioned, there is the synchronization of the cells built in this form. The selection of the cell in upper sub-form automatically makes the selection of the cell at the same position in the lower sub-form. It is shown on the figure Fig 7.

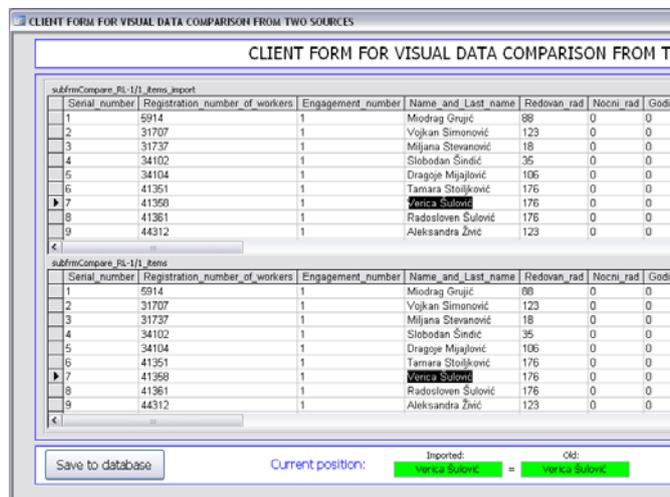


Fig 7. Synchronized navigation of the cells in two sub-forms

Both sub-forms in the form are formatted to have the same number of rows and columns. When user selects a cell in any of sub-forms, appropriate events trigger VBA source code to select the cell on the same position in another sub-form. The event property used in VBA is SetFocus and appropriate source code is shown at Fig 8.

```
Private Sub Name_and_Last_name_GotFocus()
    If CurrentProject.AllForms("frmData_Comp").IsLoaded Then
        Me.Parent.[subfrmCompare_RL-1/1_items_import].Form.Name_and_Last_name.SetFocus
    End If
End Sub
```

Fig 8. The VBA source code of event SetFocus that is used for synchronization of the cells in both sub-forms

At the end of visual comparison process, the user can save modified data to database and export data in predefined Excel file [10].

Our software application is in production testing phase in one of the clients in Secondary School in Kursumlija, Serbia.

6. CONCLUSION

In the manual preparation of the file for batch input to the centralized database system, the source data might be recorded on separate files. The files can be gathered in the same directory, but the analysis of the data in the different files cannot be performed automatically. Also there must be implemented the naming convention of the files in order to keep track on file versions. The solution presented in this paper enables to the clients of centralized database system to prepare data for batch input easily and to keep track of all entered source data.

If all source data are entered in client database on time, the number of errors is small and there is a lower probability that some data will be missed. All entered source data is at one place and any kind of analysis and processing can be performed.

Because all human can make mistakes during the data input, before the final version of data is sent to the centralized database system, client can compare own prepared data with expected data generated by centralized system. When centralized system provides initial data in interface file to the client, client can do control of own data against the expected one. As the format of interface file is pre-defined, the data prepared from the client can be compared to the data sent from the centralized system in this – pre-defined format. All deviation of values between two sources are visually marked in red color, so the client can easily compare data and after eventual correction of own data, export final data to pre-defined interface file.

We expect that our solution will find wide application as soon as we finish with testing in production environment.

REFERENCES

- [1] S.S.Ilic and S. Obradovic, *Efficient validation of exceeded limit values in batch data loading by database trigger*, International scientific conference Unitech Gabrovo, Bulgaria, November 2009.
- [2] W. Chen, J. Chun, N. Ngan, R. Ranjan and M.K. Sardana *DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET*, IBM.com/redbooks, August 2006.
- [3] Jason Price, *Oracle Database 10g SQL*, McGraw-Hill/Osborne © 2004.
- [4] Joseph Sack, *SQL Server 2008 Transact-SQL Recipes*, Apress © 2008.
- [5] M. MacDonald and M. Szpuszta, *Pro ASP.NET 3.5 in C# 2008, Second Edition*, Apress © 2007.
- [6] Tim Downey, *Web development with Java – Using Hibernate, JSPs, and Servlets* © Springer-Verlag London Limited 2007.
- [7] Stevan Roman, *Access Database Design And Programming 3rd Edition – Creating Programmable Database Applications with Access 97, 2000 and 2002*, O'Reilly & Associates, Inc 2002.
- [8] Synchronize Multiple SubForms, http://www.rogersaccesslibrary.com/forum/forum_posts.asp?TID=22
- [9] Extending the Functionality of Conditional Formatting in Access, [http://msdn.microsoft.com/en-us/library/aa139965\(office.10\).aspx](http://msdn.microsoft.com/en-us/library/aa139965(office.10).aspx)
- [10] Database solutions and downloads for Microsoft Access <http://www.databasedev.co.uk/>