Technical Report on Distributed Collaborative Applications for Econometrics

Cosmin TOMOZEI
PhD Candidate, University of Economics Bucharest
Assistant-Lecturer, “Vasile Alecsandri” University of Bacau
cosmin.tomozei@ub.ro

Abstract: *The objective of the technical report is to reflect on the development process of econometric software. Consequently, we will present a software application that has the role of determining the coefficients of a linear model, based on the ordinary least squares regression analysis technique. We will be focused both of the following directions, technical - computer science related perspective and also on the econometric and mathematical approach. By applying the appropriate development environments, costs are being reduced considerably and the efficiency level is appreciably augmenting. In addition to that, reengineering the already existing object oriented desktop applications brings us to the conclusion that econometric software is efficient in legacy. Consequently, the application has been transformed by means of reengineering and gained the characteristic of distribution.*

Keywords: *collaborative econometric applications, reengineering, ordinary least squares, regression analysis.*

1. Preliminary aspects regarding the regression analysis

We will begin our approach by referring to [1] and [2] where exogenous variables are described as being uncorrelated with the error term in the regression model. As a matter of fact, if we consider the following equation, the determination of the coefficients of exogenous variables would become straightforward.

Let the multi linear regression model [5] be described as the following equation:

$$y_i = \beta_0 + \beta_1 x_1 + ... + \beta_i x_i + ... \beta_n x_n + \varepsilon_i \quad (1)$$

which can be also written as:

$$y_i = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \varepsilon_i \quad (2)$$

where:

- y_i represents the dependent, explained variable;
- x_i describes the independent variables, explanatory variables;
- ε_i is the error term.
- β_i is the coefficient of the independent variable *i*, which we will have to determine in the model.

Our purpose is to estimate the coefficients of the explanatory variables, and the most frequent way for doing that is by using the ordinary least squares technique.

As a preliminary condition, we will presume that the expectation of the variable ε is 0, taking into account that *x* and ε are random variables. In other words, $E(\varepsilon|x)=E(\varepsilon)=0$, meaning that the value of ε does not depend of the values of x.

Consequently, it is straightforward to describe in the following equation the regression model without taking into account the variable ε. This will lead to the idea that:

$$E(y|x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i \quad (3)$$

In [4] is shown that the regression model may be described also by using matrices. In practice, when we build econometric software this is the most frequent algorithm:

$$X'y = X'X\beta + X'\varepsilon = X'Xb_{ols} \quad (4)$$

where:

- X represents the matrix of exogenous or independent variables;
- y is the vector of endogenous or dependent variables;
- β is the vector of coefficients which have to be estimated;
- ε represents the error term;
- X' is the transpose matrix of X.

It is straightforward to describe (4) as the following formula:

$$\begin{bmatrix} x_{11} & x_{12}\ldots\ldots & x_{1n} \\ x_{21} & x_{22}\ldots\ldots & x_{2n} \\ \ldots. & \ldots. & \ldots. \\ x_{m1} & x_{m2} & x_{mn} \end{bmatrix}' * \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_m \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12}\ldots\ldots & x_{1n} \\ x_{21} & x_{22}\ldots\ldots & x_{2n} \\ \ldots. & \ldots. & \ldots. \\ x_{m1} & x_{m2} & x_{mn} \end{bmatrix}' ** \begin{bmatrix} x_{11} & x_{12}\ldots\ldots & x_{1n} \\ x_{21} & x_{22}\ldots\ldots & x_{2n} \\ \ldots. & \ldots. & \ldots. \\ x_{m1} & x_{m2} & x_{mn} \end{bmatrix} * [\beta_1, \beta_2, \ldots, \beta_n] + \begin{bmatrix} x_{11} & x_{12}\ldots\ldots & x_{1n} \\ x_{21} & x_{22}\ldots\ldots & x_{2n} \\ \ldots. & \ldots. & \ldots. \\ x_{m1} & x_{m2} & x_{mn} \end{bmatrix}' * \begin{bmatrix} \varepsilon_1 \\ \ldots \\ \varepsilon_m \end{bmatrix}$$

(5)

The estimation of β by the ordinary least squares method will become [3] [4]:

$$\beta_{\text{ols}} = (X'*X)^{-1} * X^T * y \quad (6)$$

This hypothesis is the basis of the following representation and software analysis. β_{ols} represents the estimated coefficients of the linear model based on the ordinary least squares.

2. UML Representation of collaborative econometric software

The subsequent diagrams show how the system has evolved after the process of reengineering [6]. It has become a distributed application with machine readable web services developed on the .NET platform.

In Figure 1 the class diagram before reengineering is presented. In Figure 3, we will show as well the sequence diagram in order to provide a dynamic image of the functionalities.

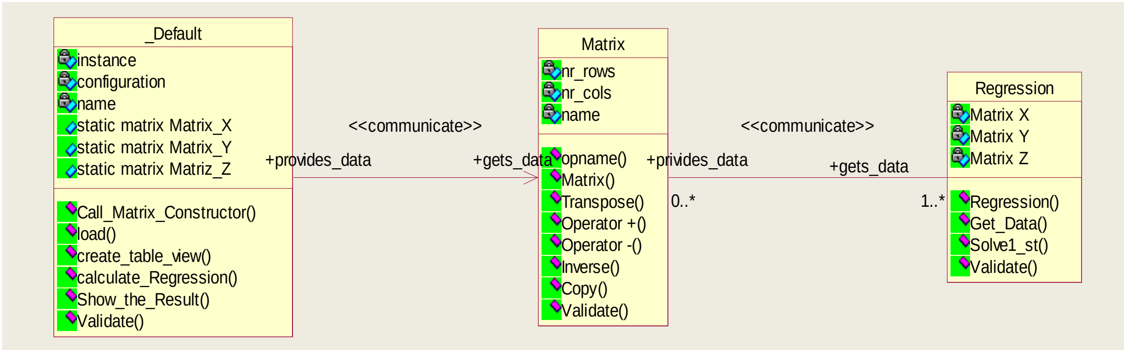

Fig. 1. Class diagram for the regression model before reengineering

After the process of reengineering, the application becomes web oriented, with dynamic content and highly opened, reliable and scalable. In Figure 2, we present the class diagram after the process of reengineering. The metrics of complexity, reliability and maintainability are to be briefly presented on behalf of the correlation of the theoretical approaches with our practical results.

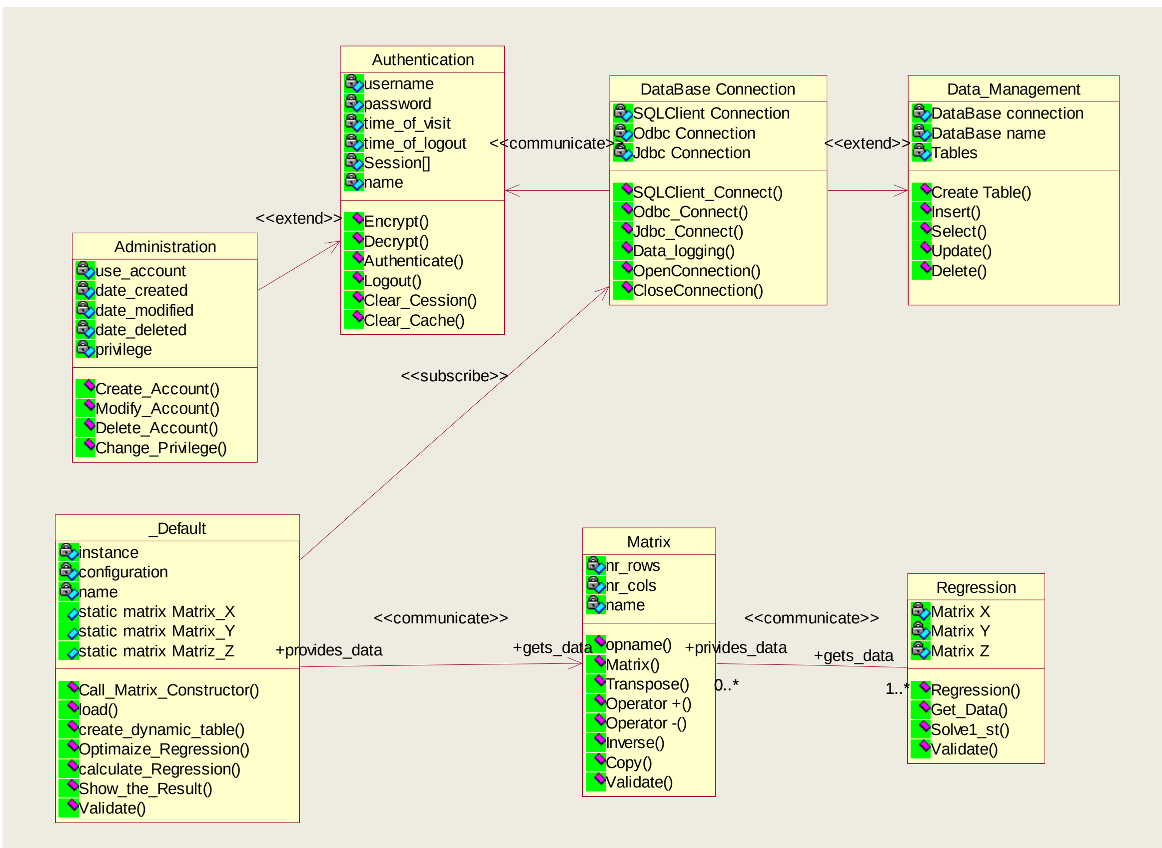

Fig. 2. Class diagram for the regression model after reengineering

As it is shown in the class diagram, new functionalities have been added, web orientation has been achieved, and a dynamic approach has been implemented. It is very important for distributed econometric application to be collaborative. Administration and authentication modules have to be included. They offer a clear perspective about each individual client's [7] [8] [9], cooperation with the system.

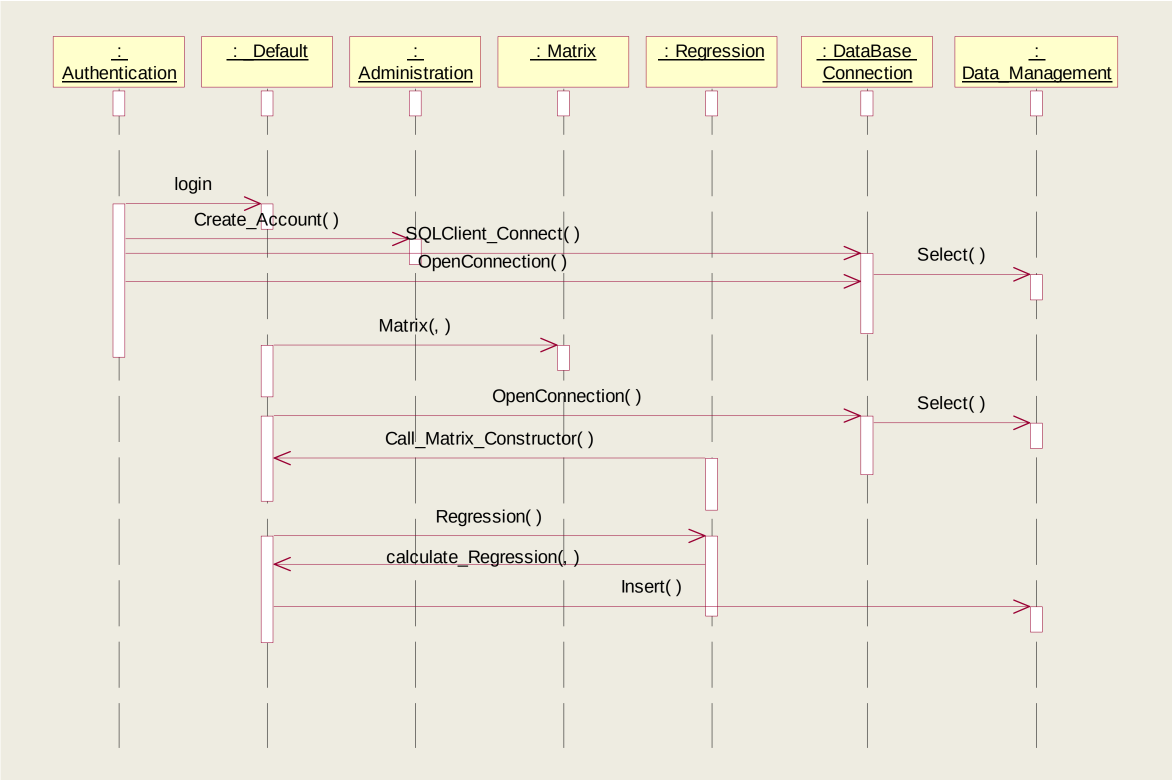

Fig. 3. Sequence diagram for distributed econometric OLS application

In [1] [7] and [9] it is mentioned that Collaborative systems must work better than other types of systems. An example of collaborative information system, defined as a distribution company whose objective is to sell increasing quantities of its products is presented. The econometric applications should respect this condition too, and offer a higher degree of usability, reliability and robustness in balance with the other types of software applications.

Quality characteristics are in expressed in quantitative way by metrics. As in [1], we put into operation the following metrics, which we have redesigned for econometric applications:

- the project *complexity* has been be reduced, so as the application to respond rapidly to the demands and not to need much resources; the reengineering process did not increased the level of complexity; we believe the positive independence [10] characteristic has an impact on the project complexity in collaborative systems;
- *reliability* of the project assumes that sometimes failures in business or econometric software applications are causing bad consequences; a distributed application may also be depending of the collection of hardware, software and communication elements that serve the objectives. An Internet application must work 24 hours a day, seven days a week without any problems that may generate failures in the system. Testing, management and reengineering greatly affect reliability. It is advisable to test all the modules that are affected by reengineering and make statistic determinations of the results;
- maintainability [1] presumes that from the moment we created the new software, it would become easy to adapt it to new demands, new estimation models and new functionalities. Furthermore new objectives are to be realized.

All these metrics have been considered during the development of the application.

3. Practical Results in programming

Here are presented some code parts from our econometric application, in which we estimate the coefficients of a linear model, described in our first part.

The code which originally worked only on desktop applications, without any possibility of being accessed by the Internet, has now evolved and easy to use and integrate by other software application, by means of SOAP. Consequently, each of the methods has also a web method correspondence in web services.

To begin with, it was necessary to read the UML class and sequence diagrams in order to have a clear view about how the classes have to be developed and how they interact.

```
public partial class _Default : System.Web.UI.Page
{
    #region matrice_statice_dinamice
        public static Matrice matrice_integrala_x;
        public static Matrice matrice_integrala_y;
        public static Matrice matrice_integrala_xy;
        public static Matrice  ygrec;
        public static Matrice rezultat;
        public static Matrice rezultat_2St;
    #endregion

    #region nr_linii_coloane

        public int nr_linii = 0;
        public int nr_coloane1= 0;
        public int nr_coloane= 0;
        public string @datefis = DateTime.Now.ToShortTimeString();

    #endregion

    protected void Page_Load(object sender, EventArgs e)
    {
        genereaza_tabel(nr_linii ,nr_coloane, nr_coloane1 );

    }

private void genereaza_tabel(int nr_linii,int nr_coloane, int nr_coloane1)
     {
System.Web.UI.WebControls.Label[,] lblxtb = new
System.Web.UI.WebControls.Label[nr_linii, nr_coloane1 + nr_coloane];

primtabel.BorderStyle = System.Web.UI.WebControls.BorderStyle.Outset;
primtabel.BackColor = Color.Gray;

        for (int I = 0; I < nr_linii; i++)
         {
             TableRow tbr = new TableRow();
             primtabel.Rows.Add(tbr);

             for (int j = 0; j < nr_coloane + nr_coloane1; j++)
             {
                 TableCell tbc = new TableCell();
                 tbc.Width = 120;
                 TextBox txtb = new TextBox();
```

```
                lblxtb[I, j] = new System.Web.UI.WebControls.Label();
                txtb.Width = 60;
                lblxtb[I, j].Width = 50;
                txtb.Text = "0.0";
                txtb.ID = "TextBoxRow_" + i+ "Col_"+ j;
                tbc.Controls.Add(lblxtb[I, j]);
                tbc.Controls.Add(txtb);
                if (j < nr_coloane)
                {
                    lblxtb[I, j].Text = "Y" + I + "," + j;
                    txtb.BackColor = Color.FromArgb(127, 255, 0, 0);
                }
                else
                {
                    lblxtb[I, j].Text = "X" + I + "," + (j - nr_coloane);
                }
                tbr.Cells.Add(tbc);
            }
            primtabel.Rows.Add(tbr);
        }
       PlaceHolder1.Controls.Add(primtabel);
    }

protected void btnCalculeaza_Click(object sender, EventArgs e)
     {
         TextBox4.Visible = true;
         Label5.Visible = true;
         try
         {

      TextBox4.Text += "Rezultat prin 1 stage least square";

      rezultat = new Matrice(int.Parse(ViewState["rows"].ToString()), 1);
      rezultat = Regresie.Rez_1st(matrice_integrala_x, matrice_integrala_y);

        for (int I = 0; I < rezultat.nrlinii; /* rezultat.nrlinii*/ i++)
           {
            for (int j = 0; j < rezultat.nrcols; j++)
                {
             TextBox4.Text += "\nBeta" + I + " 
rezultat.elem[i,j].ToString();
                }
                TextBox4.Text += "\n";
            }
         }
         catch (NullReferenceException ex)
         {
             Response.Write(ex.Message + « Apasati citeste Matrice ») ;
         }
   }
```

The *_Default* class has the role of instantiating the matrix and regression objects, in order to make the necessary calculations. Still, when working in the default class, it is compulsory to consider the matrix objects as static for avoiding errors and misunderstandings.

The regression equation is implemented in the *Rez_1St* function from the Regression Class. The name *Rez_1St* comes as an abbreviation from the ordinary least squares, or just least squares estimation technique.

The matrix class presumes the regular attributes and operation as in the traditional way of working with matrices. Reengineering process optimized the class, eliminated the unwanted redundancy and gave the opportunity for the matrices to by dynamically handled. Static attributes have been put to the matrices transmitted in the *_Default* class, with the purpose of transmitting to the compiler that the *_Default* class is not entitled to create objects of the matrix type, but to use the already defined type in the class. Consequently, the matrix attributes are included in the *_Default* class where instantiated are also, but they do not belong to any other objects.

There is also possible to estimate the coefficients of a linear regression model also by means of two-stage least squares estimation technique, when dealing with instrumental variables.

The application also has this possibility, which will be described in a future paper, as continuation of the work.

4. Conclusions and future work

We intended to create a distributed econometric application by software reengineering. Estimating the coefficients of linear models is very important in practice. The software application gained the collaborative side, by opening it through the Internet, and making it available for many categories of users.

The econometric regression model which is based on the ordinary least squares method is significantly used in different areas of research, from market research to education. Reengineering made the process of software development efficient, it saved a lot of time and reduced significantly the costs.

We would prefer to continue this research by adding to the present package of applications some new models, like the two-stage least squares estimation technique and polynomial regression.

Testing on the two-stage package has been made, and the code part is ready to be integrated in the application. A paper about the estimation in two-stage least squares method is to be published as well.

References

[1] I. Ivan, C. Ciurea and D. Milodin, “Collaborative Educational System Analysis and Assessment,” *Proceedings on The Third International Conference on Advances in Computer-Human Interactions, ACHI 2010*, Saint Maarten, Netherlands Antilles, 2010.

[2] I. Ivan, M. Doinea and D. Palaghiță, “Optimization of authentication processes in distributed applications,” *Theoretical and Applied Economics*, No.6, June 2008, pg. 39 – 56.

[3] J. Johnston, *Econometric Methods*, The McGraw – Hill Companies, 2000.

[4] D. L. McFadden, *Course Material, Chapter 4. Instrumental Variables*, Berkley University, Available at: http://elsa.berkeley.edu/~mcfadden/e240b_f01/ch4.pdf

[5] J. M. Wooldrige, *Introductory Econometrics, A Modern Approach*, pp.506-546, South Western Cengage Learning, 2009.

[6] S. Demeyer, S. Ducasse and O. Nierstrasz, “Object-Oriented Reengineering Patterns,” *Elsevier Science*, pp. 147-268., Square Bracket Associates, 2008.

[7] C. Tomozei, „Distributed Collaborative Software Solution for Higher Education,” *The Ninth International Conference of Informatics in Economy*, ASE Bucharest, 2009, pp. 113-119.

[8] C. Tomozei amd M. Vetrici, “F# Modules Integration in Distributed Applications Development Process,” *Scientific Studies and Research Series Mathematics and Informatics*, Vol. 19, No. 2, 2009, pp. 459 – 470.

[9] C. Tomozei and F. Floria, “Questions regarding alterity in social collaborative networks,” *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, Vol. 1, No. 1, 2009, pp. 70 – 75.

[10] B. Becker and G. Mark, *Constructing Social Systems through Computer-Mediated Communication*, Springer-Verlag London ltd Virtual Reality, 1999, pp. 4:60-73.

Author

Cosmin TOMOZEI is University Assistant - Lecturer at Mathematics and Computer Science Department from Faculty of Sciences of the “Vasile Alecsandri” University of Bacau. He is a PhD candidate from October 2007 at Economic Informatics Department from University of Economics, Bucharest. He holds a Master in Science - Databases- Business Support from University of Economics, Bucharest. He graduated in Economic Informatics at Faculty of Economic Cybernetics, Statistics and Informatics in 2006. His main research areas are: object oriented programming, functional programming in Lisp and F#, software reengineering and distributed applications development.