Journal of
Applied
Business
Information
Systems

http://www.jabis.ro

Metrics Implementation for Collaborative Systems

Cristian CIUREA*

* Economic Informatics Department, Academy of Economic Studies, Bucharest, Romania

Abstract: *The paper presents the main types of collaborative systems encountered in the economic field. The quality characteristics of these collaborative systems are described and methods for building and validating metrics are developed. A genetic algorithm was implemented in order to determine the local maximum and minimum points of the relative complexity function. Metrics are used in the process of collaborative systems reengineering.*
Keywords: *metrics, collaborative systems, genetic algorithm, quality characteristics.*

1. Types of collaborative systems in the economic field

Collaboration means more than two agents working together. It requires defining a shared goal and, in order to achieve this goal, the agents should create an agreement upon their ways of actions. Such an agreement is only achievable through negotiation [1].

Collaborative systems are an important research field of knowledge-based society and many human activities are involved in this area. Science has great impact on the development of different types of collaborative systems, classified by followings criteria: level of complexity, field of application and manner of organization.

Using the field of application criteria, collaborative systems are classified into several categories:

- *collaborative educational systems*, which are applied in the educational field and have the objective to evaluate and increase the performance of the educational process;
- *collaborative banking systems*, which are encountered in banking field and are used by various financial units;
- *collaborative systems of defense*, that are encountered in military field and are defined by strict rules of organizing and functioning;
- *collaborative systems in production*, their objective being to increase production capabilities and product quality within different goods and services production units;
- *collaborative functional systems*, refers to all the activities taking place in the economy, providing necessary information and overall coordination for production and finance management;
- *collaborative micropayment systems*, that allows customers and content providers to use their payment system of choice;
- *collaborative planning systems*, which present the most appropriate way to tackle certain kind of planning problems, especially those where a centralized solving is unfeasible;
- *collaborative tagging systems*, which provide a new means of organizing and sharing resources;
- *collaborative writing systems*, their major benefits include reducing task completion time, reducing errors, getting different viewpoints and skills, and obtaining an accurate text;
- *collaborative medical systems*, in which modern communication technologies allow doctors from around the world to work on the same patient, in the same time [2].

The collaborative system from the economic field works under the black box principle set out by Zadeh, the entries being given by raw materials and information and the outputs being materialized in finished products, services and other information which turns into costs for that business.

2. Quality characteristics of collaborative systems

The collaborative system is developed based on a set of specifications that were defined in the analysis stage in order to define objectives for the development process. The system must behave and must give the results the users want and that they have stated at the start.

The *complexity* is a measure for the interdependencies between components and their links and also for the diversity of different types of input and output constructions. This characteristic describes the density of fluxes between the components of the system. The complexity of the collaborative system generates a large number of various components. Based on that, a proper approach of the system quality is to analyze every component separately.

The system *reliability* is determined by analyzing the number of problems solved by the system and the total number of specified problems.

The *maintainability* is a process particular to software products that have a complex development process and that are intended to be used for a long time, meaning more than three years. In this category are included also products like the collaborative systems. Maintainability measures the effort needed to make modifications on the collaborative system in order to make it suited for current needs. This effort can be described as consumed time, number of modules modified, number of added modules and number of deleted modules [3].

The system *functionality* describes a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.

In the case of collaborative systems, the *orthogonality* finds its applicability in studying the level of similitude of the information processed within the systems and in establishing the way in which applications and implemented technologies covers the demand of presenting and adapting the information [4].

Another quality characteristic is the *usability* of collaborative systems, defined by the ability of a system to be useful for his agents. Usability of a collaborative system is reflected through the effective interactions between its agents and the successful achievement of proposed objectives.

Each quality characteristic analyzed has an indicator associated. This indicator can lead to choosing the most efficient level. The identification of this metric has a great importance in a direct comparison of the quality characteristics levels for two or more collaborative systems.

3. Techniques for building and validating metrics

A metric of collaborative systems is a mathematical model developed around an equation. Metrics use analytical expressions having the form: $y = f(x, z, w)$, where x, z and w are variables of influence factors, and y is the result variable.

The technology for building metrics of collaborative systems has the entry point that collaborative systems are influenced by the factors $f_1, f_2, \ldots, f_n$, which have associated a series of variables. It is realized a graph of influences and is made a factorial analysis, in order to analyze the variables. There is built a correlation between these influence factors. Result a lot of dependent variables and a lot of independent variables. There are built analytical expressions. It is specified the performance criterion and result a technology that allow building the optimal metric in relation to a lot of influence factors and a performance criterion.

This technology must be associated with the quality characteristic for which will be determined the metric of collaborative system.

The metrics must be not too complicated because it will use lots of resources when implemented and also it must be not too simple because the measured levels will loose relevance.

A metric of collaborative systems must be characterized by the following properties: sensitivity, not compensatory character, not catastrophic character, representativeness.

In order to measure the metric representativeness, the following indicator is used:

$$R = \frac{Dc}{Dt},$$

where:

Dc – correct decision number based on indicator value;

Dt – decision total number taken in analysis process of the collaborative system on the base of the indicator taken into account.

The *R* indicator is a relative one and it measures the degree in which the values obtained through metric applying represented a support in decisional process.

The value of the indicator *R* is included in the interval *[0; 1]*. The indicator has a maximum representativeness degree for a value of $R = 1$.

In order to measure the degree of knowledge management that a collaborative system has, is defined a knowledge management performance indicator, KMPI, as follows:

$$KMPI = \frac{KC + KA + KS + KU + KI}{5},$$

where:

KC - the knowledge creation indicator in a collaborative system;

KA - the knowledge accumulation indicator in a collaborative system;

KS - the knowledge sharing indicator in a collaborative system;

KU - the knowledge utilization indicator in a collaborative system;

KI - the knowledge internalization indicator in a collaborative system.

The quality of a collaborative system is defined as all features and characteristics, bearing ability to meet the needs specified or implied. To measure the quality of a collaborative system and assess its performance is used the indicator:

$$Q = p_1 * \frac{\min(x, y)}{\max(x, y)} + p_2 * \frac{\min(z, w)}{\max(z, w)},$$

where:

x, z – the planned values for two quality characteristics;

y, w – the realized values for two quality characteristics;

p_1, p_2 – the share of each quality characteristic ($p_1 + p_2 = 1$).

For the shares $p_1 = 0.4$ and $p_2 = 0.6$, the realized value $y = 75$ and the planned value $x = 80$, for the first quality characteristic, and for the realized value $w = 95$ and the planned value $z = 100$, for the second quality characteristic, the quality indicator is $Q = 0.945$. A value of *Q*, which is very near to *1*, means that the collaborative systems meet very well the specified criteria.

Some experimental results and their diagram are presented in the Figure 1:

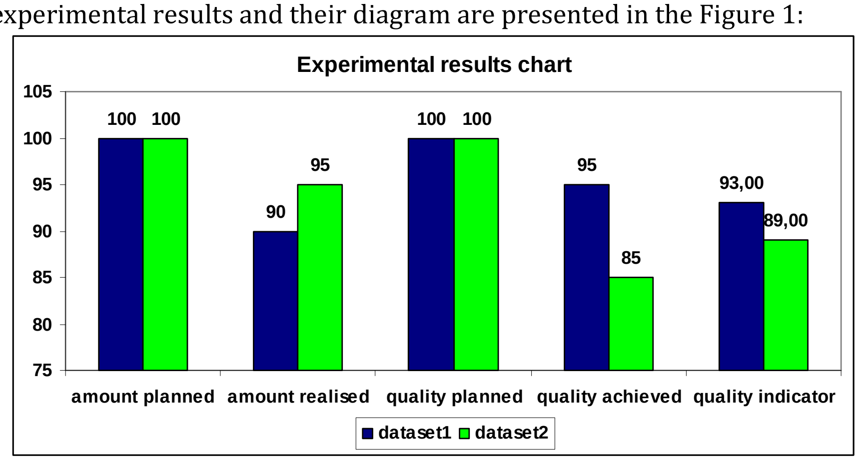

Fig. 1. Experimental results chart [3]

The current database contains a representative number of records relating to the behavior of a banking system and accepts extensions for other collaborative systems.

For the same amount and quality planned in the first dataset, when the amount realized is *90%* and the quality achieved is *95%*, the quality indicator is *0.93*. In the second dataset, for the same amount and quality planned, when the amount realized is *95%* and the quality achieved is *85%*, the quality indicator has the value *0.89*.

This quality is achieved at the end of the developing process of collaborative systems, if, during the development, are built those internal properties that determine the level of quality characteristics.

4. Defining a genetic algorithm for metrics implementation in a collaborative banking system

Collaborative systems differ one from each other by complexity. The complexity problem is made similarly to the problem of simplicity. The complexity of collaborative systems is a new concept that requires a rigorous definition in order to measure the level of complexity and to compare the systems.

The optimal complexity of a banking information system, *CO*, is determined according to relationship [5]:

$$CO = \sum_{i=1}^{n} x_i * \log_2 x_i ,$$

where:

x_i – the number of components associated with the i^{th} software application of the banking information system, with the property that $x_i > 0$ are natural numbers, $x_i \in N$.

Figure 1 shows the 3D graphic of the function $f(x) = x * \log_2 x$, where $x \in N$.

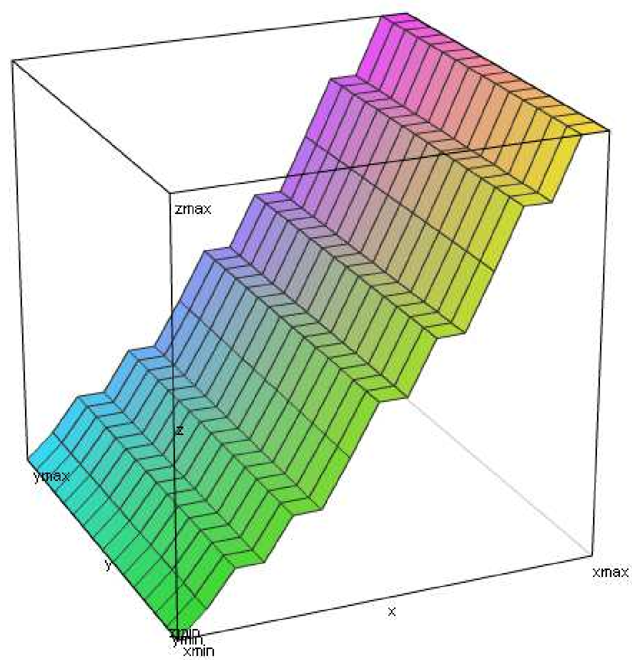

Fig. 1. The 3D graphic of the function $f(x) = x * \log_2 x$

For n=2 results that $CO = x_1 * \log_2 x_1 + x_2 * \log_2 x_2 = x * \log_2 x + y * \log_2 y$.

Figure 2 shows the 3D graphic of the function $f(x, y) = x * \log_2 x + y * \log_2 y$, where $x>0, y>0$, $x \in N, y \in N$.

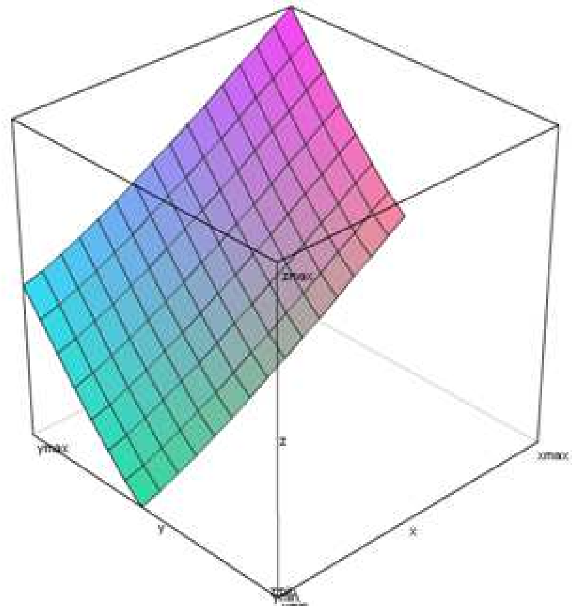

Fig. 2. The 3D graphic of the function $f(x,y) = x * \log_2 x + y * \log_2 y$

First order function derivatives of the function $f(x,y) = x * \log_2 x + y * \log_2 y$ are:

$$f'_x(x,y) = \log_2 x + \frac{1}{\ln 2},\ f'_y(x,y) = \log_2 y + \frac{1}{\ln 2}.$$

In order to determine the stationary points, the first order derivatives are equalized with zero.

To check whether the stationary point is a maximum or a minimum point, there are calculated the second order derivatives of function and Hessian matrix is built.

$$f''_x(x,y) = \frac{1}{x * \ln 2}.$$

$$f''_y(x,y) = \frac{1}{y * \ln 2}.$$

$$f''_{xy}(x,y) = 0.$$

Because $x>0$ and $y>0$, the first order minor of Hessian matrix is positive defined as the minor of second-order.

The relative complexity of a banking information system, *CR*, is determined according to the relationship:

$$CR = \frac{\sum_{i=1}^{n} x_i * \log_2 x_i}{\sum_{i=1}^{n} x_i * \log_2 \sum_{i=1}^{n} x_i} = \sum_{i=1}^{n} \frac{x_i * \log_2 x_i}{\sum_{i=1}^{n} x_i * \log_2 \sum_{i=1}^{n} x_i},$$

where:

x_i – the number of components associated with the i^{th} software application of the banking information system, with the property that $x_i > 0$ are natural numbers, $x_i \in N$.

In Figure 3, the 3D graphic of *CR* complexity function is presented:

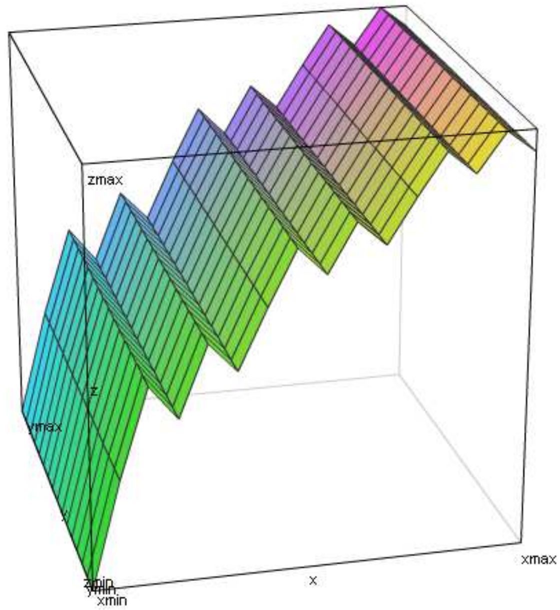

Fig. 3. The 3D graphic of the function *CR*

Figure 4 shows the same 3D graphic of the function *CR*, but from different points of view:

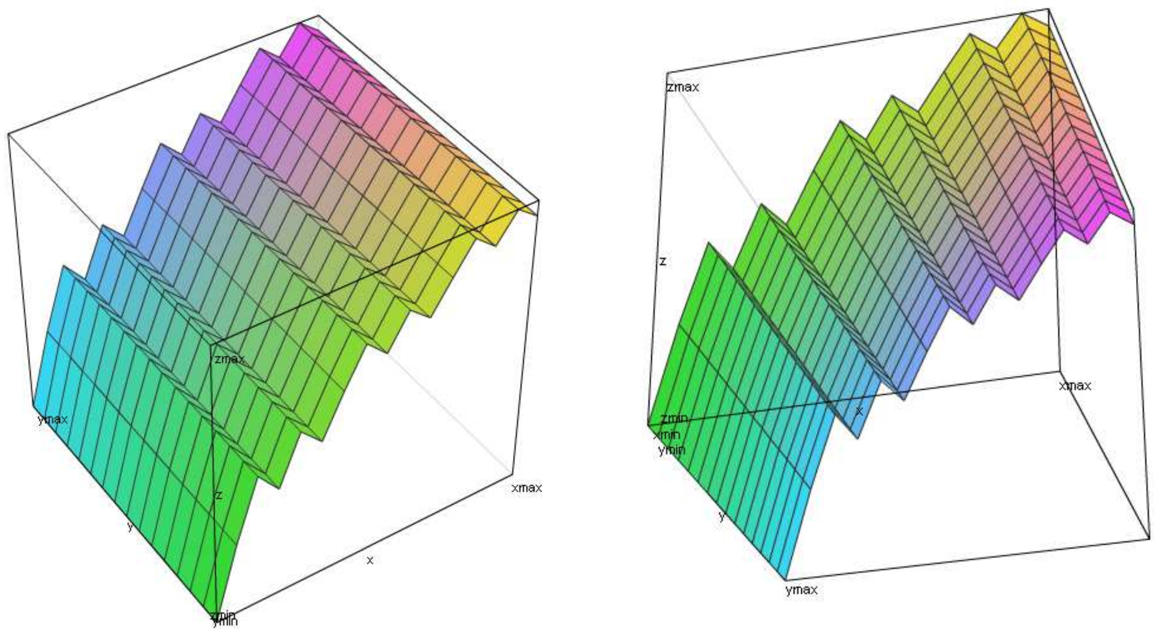

Fig. 4. The 3D graphic of the function *CR*

Let consider $g_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$. Results that $CR = \sum_{i=1}^{n} g_i * \frac{\log_2 f_i}{\log_2 \sum_{i=1}^{n} f_i}$.

For n=1, results *CR=1*.

For n=2, results $CR = \frac{f_1 * \log_2 f_1 + f_2 * \log_2 f_2}{(f_1 + f_2) * \log_2 (f_1 + f_2)} = \frac{x * \log_2 x + y * \log_2 y}{(x + y) * \log_2 (x + y)}$.

In Figure 5 the 3D graphic of the function $f(x, y) = \frac{x * \log_2 x + y * \log_2 y}{(x + y) * \log_2 (x + y)}$ is presented:

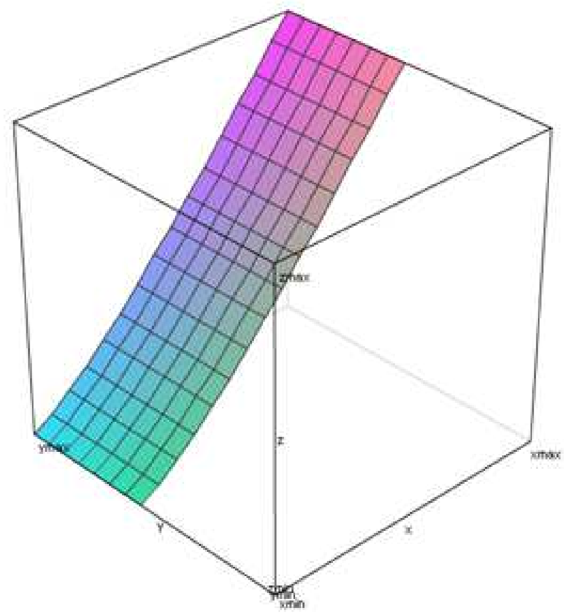

Fig. 5. The 3D graphic of the function $f(x, y) = \dfrac{x * \log_2 x + y * \log_2 y}{(x + y) * \log_2 (x + y)}$

In order to determine the local minimum and maximum values of the *CR* function, where $x_i \in R$, a genetic algorithm has been implemented within Collaborative Multicash Servicedesk application. This algorithm objective is to determine the number of components of the banking information system for which the relative complexity is minimum or maximum.

The source code was written in C# programming language, being implemented the classes *GeneticAlgorithm*, *Genome* and *GenomeComparer*, as follows [6]:

```
public delegate double GeneticAlgorithmAFunction(double[] values);

public class GeneticAlgorithm
{
 static private GeneticAlgorithmFunction getFitness;
 public GeneticAlgorithmFunction FitnessFunction
 {
  // etc.

 };
 // etc.
}

GeneticAlgorithm ga = new GeneticAlgorithm(0.8,0.05,100,2000,2);

ga.FitnessFunction = new GeneticAlgorithmFunction(theActualFunction);

public sealed class GenomeComparer : IComparer
{
   public GenomeComparer()
   {
   }
   public int Compare( object x, object y)
   {
      if ( !(x is Genome) || !(y is Genome))
         throw new ArgumentException("Not of type Genome");
```

```
    if (((Genome) x).Fitness > ((Genome) y).Fitness)
      return 1;
    else if (((Genome) x).Fitness == ((Genome) y).Fitness)
      return 0;
    else
      return -1;
  }
}
```

The *Genome* class was built as a simple container. A matrix whose elements lie in the range 0-1 gives the basic structure. The algorithm will use these values, and the user will expand them to the scale of needs. Because mutations occur on the genome, in the *Genome* class there is the *Mutate()* method, implemented as follows:

```
public void Mutate()
{
for (int pos = 0; pos < m_length; pos++)
{
if (m_random.NextDouble() < m_mutationRate)
m_genes[pos] = (m_genes[pos] + m_random.NextDouble()) / 2.0;
}
}
```

The *Crossover()* method needs access to private data of the genome, so it is a class member function in the *Genome* class, this method exits being two child objects of the *Genome* class.

```
public void Crossover(ref Genome genome2, out Genome child1, out Genome child2)
{
int pos = (int)(m_random.NextDouble() * (double)m_length);
child1 = new Genome(m_length, false);
child2 = new Genome(m_length, false);
for (int i = 0; i < m_length; i++)
{
if (i < pos)
{
child1.m_genes[i] = m_genes[i];
child2.m_genes[i] = genome2.m_genes[i];
}
else
{
child1.m_genes[i] = genome2.m_genes[i];
child2.m_genes[i] = m_genes[i];
}
}
}
```

The genetic algorithm requires the following steps:

- creating a new population;
- selecting the best two individuals from the population and cross them for obtaining children;
- replacing the old population with a new one;
- resumption of the previous steps until it reaches the optimal solution of the problem.

Figure 6 shows how to calculate the extreme points of local maximum and minimum of *CR* function using the genetic algorithm implemented within the CMS application:

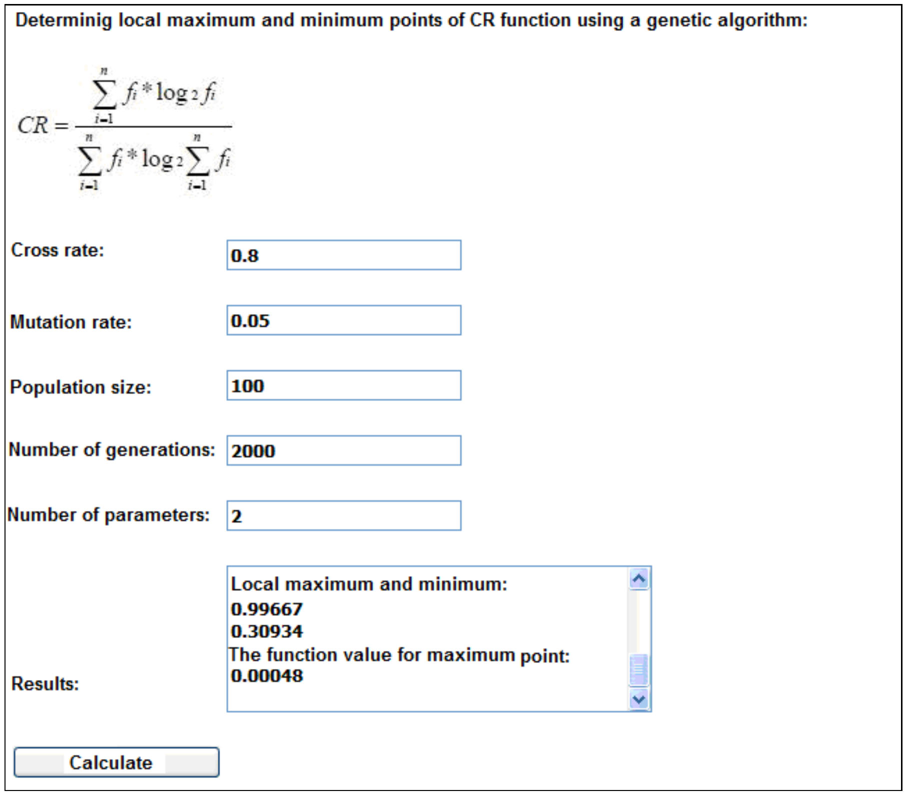

Fig. 6. Determining the local maximum and minimum of CR function

At the first call of the genetic algorithm, the local maximum and minimum were the followings:
The local maximum and minimum points:
0.9476
0.68387
The function value in the maximum point = -0.00785

At the next call, the values were improved, being closer to real values:
The local maximum and minimum points:
0.98459
0.55019
The function value in the maximum point = -0.00225

In the third call, the values obtained were much closer to the maximum and minimum points:
The local maximum and minimum points:
0.99667
0.30934
The function value in the maximum point = -0.00048

The local minimum point value, i.e. *0.30934*, shows that both optimal and relative complexities are minimal for $x_i = \frac{1}{e}, i = 1..n$, $x_i \in R$. This is the number of components for which the relative complexity of the banking information system is minimum.

5. Using metrics for collaborative banking systems reengineering

In the collaborative banking systems the following components are encountered:

- *the material*, which includes buildings, equipment and other property;
- *the energy*, consists of flow of electricity, Internet and intranet connections, alternative channels of communication;

- *the information*, comprising all software and hardware resources available to the bank to conduct its business;
- *the human*, including the bank human resources, the categories of personnel and the qualification levels of them.

Bank's financial results depend largely on the quality of staff and the efforts of each employee separately. For each position of the bank, the department of human resources is seeking people with a degree of training higher than required by the job in question. The goal of this recruitment is the elimination of cases in which an employee fails to meet certain requirements or to resolve certain issues related to its activity. Training of employees at work must be done at least every five years. During this period, an employee of the bank carries out one or more trainings.

Regarding the collaborative banking system, an indicator for increasing the efficiency is the level of staff training. Considering the qualifications period of five years, the minimum number of qualifications that get an employee is one in five years and the maximum number is one per year or five qualifications over five years. The maximum number of training sessions that the bank will finance, over a period of five years, is calculated according to the relationship:

$NT = 5 * NP$,

where:

NT - the total number of training or qualifications supported by the bank;

NP - the numbers of people employed in the bank and are eligible for training.

If we take into account the duration of trainings, in the formulas for calculation of the indicators will appear another two variables:

$Dmin$ - minimum duration of training, expressed in calendaristic months;

$Dmax$ - maximum duration of training, expressed in calendaristic months.

In this case, the total number of training sessions supported by the bank within five years, expressed in calendaristic months, is given by the relationship:

$NT = 5 * NP * Dmax.$

The degree of increasing the level of staff training will be determined with the same formula, with the difference that the number of persons qualified in five years is weighted with the duration of qualifications for each person.

It is considered the database of Collaborative Multicash Servicedesk - CMS application, in which are stored the requests of a bank customers, relating to the problems that they have in using the Multicash electronic payment service.

The Collaborative Multicash Servicedesk application is structured in two modules:

- the module for online registration of bank customers requests;
- the module for recording phone requests by Multicash Helpdesk analysts.

In the module for online registration of bank customers' requests, each customer receives from the bank a username and password with which he will authenticate in the application. The associated customer interface allow the customer to send a written request to the Helpdesk department, by framing the issue in the appropriate category and subcategory, but also to register a priority request in exchange of a fee.

In the module for recording phone requests by Multicash Helpdesk analysts, after authentication in the application, the analyst see the page from which is made the registration of requests in the database.

The fields to be completed or selected by the bank analyst are the followings:

- customer name, based on suggestions from a predefined list of Multicash customers;
- the contact person of the customer who made the call;
- the request category, which is a drop-down list with predefined categories and related codes;
- request description, which is a field for adding the details of the problem;
- the way to solve by selecting the appropriate option.

The CMS application is used effectively within Raiffeisen Bank, in its database is being introduced over two thousand requests per month. Having the database of all customer requests, it is realized the analysis of the types of problems faced by Multicash service users and are determined the strategies to address each customer, according to the history of problems he encountered.

Working with the entire database of requests allows avoiding future complaints through the analysis of previous customers' problems, offering solutions and additional support to the customers with many requests.

The situation of requests on categories, recorded in the period October 15 to November 15, 2010, is presented in Table 1:

Table 1. Number of requests on categories

Request code	Request category	Number of requests
901	Training on using the application	107
902	User blocked at logon	127
903	User blocked on the communication	248
904	Training on see rejected payments	56
905	Check payments status	795
906	Login with admin2 user	3
907	Index corrupted in database tables	27
908	Please repeat job with AC29	37
909	Communication initiated	254
910	Transmission interrupted	155
911	Signature error	233
912	Generate electronic signature	122
913	Add new users in the client application	95
914	Add new accounts in the client application	105
915	Change name / address of payer	15
916	Training of branches for completing annexes	52
917	Error on see statements	186
918	Delivery account statements	80
919	Delivery files for distributed signature	34
920	Move the application on another computer	70
921	Installing the application abroad	11
922	Confirm account balance	424
923	Deactivate payments file	11
924	Change communication channel	8
925	Setting print parameters	20
926	Reinstalling the application	54
928	Change number of approvals / amount limits	6
929	Error on starting the application	54
930	Statements export	20
931	Setting communication sessions	6
932	ROI or INT button disappearing from the main menu	2
933	Other requests	729
934	Delivery file with bank codes	83
935	Unresolved - BAS blocked	2
936	Check the import file structure	6
937	Check the validity of files sent for distributed signature	1
938	Payments cancellation	28
939	Change the customer status in LIVE/ TEST	3
940	Communication problems to the customer	17
941	Decryption error/ wrong communication password	7
942	Missing a bank branch	12
999	Intervention of service provider	9

Analyzing the data from Table 1, result that most requests were registered on *Check payments status* category, because the Multicash service allows viewing information on the settlement payments status and accounts balances updated every hour. Customers need the confirmation of certain payment processing at a certain time and they call the Helpdesk department to get these confirmations.

Graphical representation of the number of requests by category is given in Figure 7:

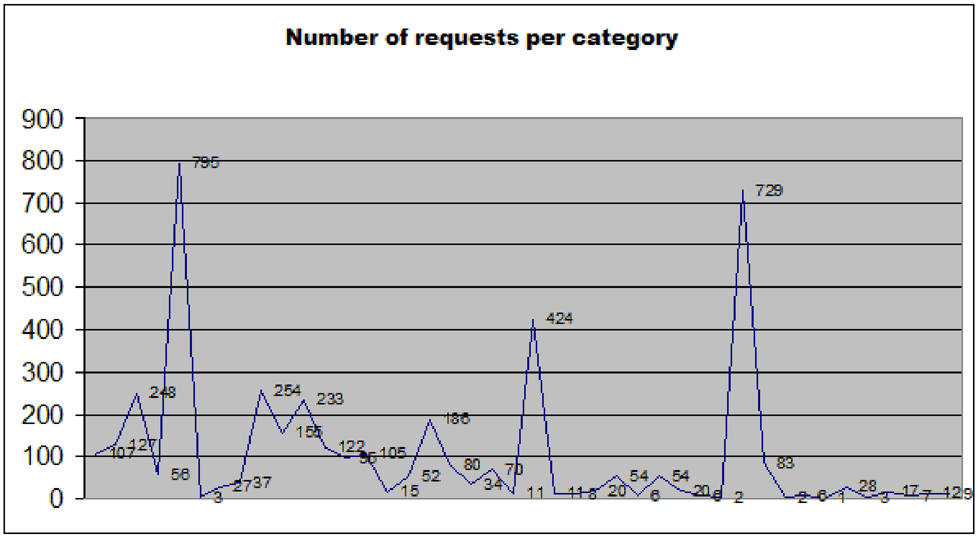

Fig. 7. The number of requests by category

According to the graphic representation in Figure 7, the first three categories with the biggest number of requests are *Checking payments status*, *Other requests* and *Confirm account balance*. The difference between the number of requests registered on these categories and the number of requests from other categories is significant. To reduce the number of requests in these categories should be:

- improved the Multicash service, in order to allow real time view of operations performed;
- reviewed the requests recorded in the category *Other requests* for their reclassification in existing categories or in order to create new categories of problems;
- updated accounts balances in real time.

Working on large data sets allow the launch of assumptions, making calculations and determining ways to correct reality.

Databases with transactions performed in a bank contains information about the user who performed the operation, the channel through was done, from which workstation, in which date and which hour. These databases are updated in real time and are consulted by the Banking Security Department to discover any fraud attempts. If you find that, from a workstation, an operator makes a lot of transactions compared to other operators, or amounts transferred are very high, then it is done thorough research regarding these operations.

From the database of CMS application data sets are identified and is performed a combined analysis to determine certain statistics. The combined analysis involves correlations between data sets, for the calculation of quality indicators.

For the analysis *Person – Operations*, are identified the types of operations made by a person.

Is determined the load degree of each agent in the system and is made a redistribution of operations so that do no exist a situation in which an agent is overloaded and another do not have enough operations which fill the working time.

It considers H_1, H_2, H_3 and H_4 the names of four analysts who actually work with the CMS application within the Multicash Helpdesk department of Raiffeisen Bank.

From the combined analysis *Analyst – Category* of requests, on the basis of records from the Collaborative Multicash Servicedesk application, results that the analyst H_1 solved requests from the categories *Add new accounts in the client application, User blocked on the communication, Generate electronic signature, Change communication channel*, and the analyst H_2 solved requests from the categories *Add new users in the client application, Training on see rejected payments, Move the* *application on another computer*. Taking into account the number of requests recorded on each category, it follows that the analyst H_1 has been overloaded.

For the analysis *Person – Resolutions*, there are evaluated the types of resolutions adopted and their frequencies of occurrence:

H_3: resolution YES at the rate of x%, NO at the rate of y%.

H_4: resolution YES at the rate of z%, NO at the rate of w%.

If $x > z$, then H_3 gave more positive resolutions than H_4. If $x > y$, then H_3 gave more positive than negative resolutions. If $z > w$, then H_4 gave more positive than negative resolutions.

In Figure 8, a logical scheme is presented in order to evaluate the situations in which an analyst gave positive or negative resolutions.

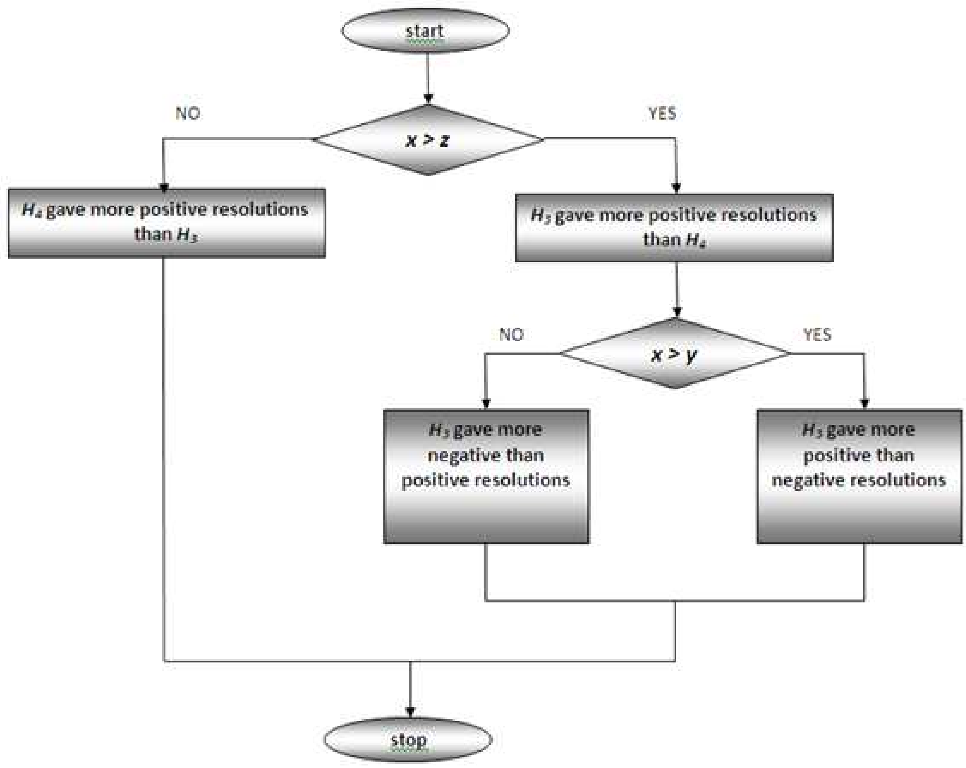

Fig. 8. The logical scheme of possible resolutions

By generalization, being considered the data sets $D_1, D_2, ..., D_n$, correlations are established between any of D_i and D_j, where $i, j = 1..n$, with $i \neq j$. For each combined analysis $D_i – D_j$ the types of correlations are analyzed and are calculated quantitative and qualitative indicators.

Indicators for the case presented above are [7]:

- the quantitative indicator comparing the number of resolutions adopted by the two entities:

$$I_{D_i/D_j} = \frac{N_{D_i}}{N_{D_j}},$$

where:

N_{D_i} – total number of resolutions adopted by D_i;

N_{Dj} – total number of resolutions adopted by D_j;

- the qualitative indicators comparing values between the two resolutions adopted:

$$I_{D_i} = \frac{x}{y};\ I_{D_j} = \frac{z}{w};$$

$$I_{x/z} = \frac{x}{z};\ I_{y/w} = \frac{y}{w};$$

For $x = 80$, $y = 20$, $z = 70$ şi $w = 30$, the indicator I_{Di} has value 4, representing the report of positive and negative resolutions established by H_3. The indicator I_{Dj} has value 2.33, representing the report of positive and negative resolutions established by H_4. The number of positive resolutions established by H_3 versus H_4 is $I_{x/z} = 1.14$, and the number of negative resolutions established by H_3 versus H_4 is $I_{y/w} = 0.66$.

The collaborative banking systems should work better than other types of systems, because these systems creates a collaborative environment where people can work better together, can share information without the constraints of time and space.

6. Conclusions

Metrics are very important in the evaluation of quality characteristics of collaborative systems. Techniques for building metrics must be applied so that the indicators built to be sensitive, not compensatory, not catastrophic and representative.

Neural networks and genetic algorithms are successfully used in metrics construction and validation. The metrics developed have a great contribution in the process of collaborative systems reengineering.

Acknowledgements

This article is a result of the project POSDRU/6/1.5/S/11 „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies, project no. 7832, Doctoral Program and PhD Students in the education research and innovation triangle, DOC-ECI.

References

[1] S. O. Oguz, A. Kucukyilmaz, T. M. Sezgin and C. Basdogan, "Haptic negotiation and role exchange for collaboration in virtual environments," *2010 IEEE Haptics Symposium*, 25-26 March 2010, pp. 371-378.

[2] P. Pocatilu and C. Ciurea, "Collaborative Systems Testing," *Journal of Applied Quantitative Methods*, Vol. 4, No. 3, 2009.

[3] I. Ivan and C. Ciurea, "Quality Characteristics of Collaborative Systems," *Proceedings of The Second International Conferences on Advances in Computer-Human Interactions, ACHI 2009*, February 1-7, 2009, Cancun, Mexico.

[4] I. Ivan, C. Ciurea and D. Milodin, "Collaborative Educational System Analysis and Assessment," *Proceedings of The Third International Conferences on Advances in Computer-Human Interactions, ACHI 2010*, February 10-16, 2010, Saint Maarten, Netherlands Antilles.

[5] I. Ivan and C. Ciurea, "Collaborative Systems Aggregation," *Journal of Applied Collaborative Systems*, Vol. 2, No. 3, 2010, ISSN 2066-7450.

[6] http://www.codeproject.com/KB/recipes/btl_ga.aspx

[7] I. Ivan, C. Ciurea and S. Pavel, "Very Large Data Volumes Analysis of Collaborative Systems with Finite Number of States," *Journal of Applied Quantitative Methods*, Vol. 5, No. 1, 2010, ISSN 1842–4562.