JOURNAL OF APPLIED QUANTITATIVE METHODS

VERY LARGE DATA VOLUMES ANALYSIS OF COLLABORATIVE SYSTEMS WITH FINITE NUMBER OF STATES[1]

Ion IVAN[2]
PhD, University Professor, Department of Computer Science in Economics,
University of Economics, Bucharest, Romania

E-mail: ionivan@ase.ro ; **Web page:** http://www.ionivan.ro

Cristian CIUREA[3]
PhD Candidate, Department of Computer Science in Economics,
University of Economics, Bucharest, Romania

E-mail: cristian.ciurea@ie.ase.ro

Sorin PAVEL[4]
PhD Candidate, Department of Computer Science in Economics,
University of Economics, Bucharest, Romania

E-mail: pavelsorin@gmail.com

Abstract: *The collaborative system with finite number of states is defined. A very large database is structured. Operations on large databases are identified. Repetitive procedures for collaborative systems operations are derived. The efficiency of such procedures is analyzed.*

Key words: *procedures, collaborative, state, database, operations, efficiency*

1. Collaborative systems with finite number of states

Within a collaborative system, a multitude of users or agents are involved in a distributed activity, most of the time being in several different places. In the large family of distributed applications, the collaborative system is identified by the common goal that the agent are working for and the great need of interaction that exists in the process of sharing and exchanging information and applications [1][5].

A software collaborative system is like a distribution company that seeks the increasing of sales. The difference between a collaborative system and a distributed one is given by the following attributes of the collaborative system:

- the system elements, both users and agents, interact with each other influencing the behavior of the system;
- the system components use shared resources in order to fulfill both their own objectives and their common goals;

- the system has permanent communication channels between users and agents;
- the agents' interests are not antagonistic (the agents have common and corporate interests).

The characteristics of collaborative systems are: complexity, reliability, portability, mentenability, structurability, stability, adaptability, operationability and integrability [3].

Types of collaborative systems counts:

- *collaborative systems in education:* active in the educational and research field and pursue increased performance and testing of the educational process;
- *collaborative systems of defense:* active in military field and are defined by strict rules of organizing and functioning;
- *collaborative systems in production:* pursuing increased production capabilities and product quality within distinct goods and services production units;
- *collaborative banking systems:* used by banks and financial units, these systems are analyzed along this paper in order to determine the parameters that influence the banking systems and all its components;
- *electronic business systems:* companies' departments are becoming more and more integrated, and clients are now users of e-business systems, thus replacing the traditional security mechanisms with authorization software – the modern security systems which mange and store users' data and correlate them with the access rules of the organization;
- *public administration systems:* used for managing tax collection, for integrated management of human resources and payroll, for querying city hall databases on citizen demand;
- *media software development systems:* media applications development was indirectly caused by the increasing of common use electronic devices; these systems include commutations stations for wireless, terrestrial, satellite and cable infrastructure.

The agents of a collaborative system interact dynamically. Therefore, the system should be flexible in ability to execute transactions. Agents, servers, data warehouses and transactions are all elements which generally compound distributed systems, but the nature of transactions between agents and shared objectives of the agents are the main parts of a collaborative system.

Let $S_1, S_2, \ldots, S_n$ be the array of states that a collaborative system cover. Changing from S_i state to S_j state is done by a message, a command, a document d_{ij}. Table 1 contains the matrix of changing by documents from one state to another for a collaborative system.

Table 1. Transition by documents between states of collaborative systems

	S_1	S_2	…	S_j	…	S_n
S_1						
S_2						
…						
S_i				d_{ij}		
…						
S_n						

The collaborative system named *bank* covers the following states:

S_1 – open;

S_2 – money receive;

S_3 – credit approve;

S_4 – money discharge;

S_5 – currency exchange;

S_6 – closed.

Table 2 contains transaction between the six states of the *bank* collaborative system:

Table 2. Transaction between the six states of the *bank* collaborative system

	S_1	**S_2**	**S_3**	**S_4**	**S_5**	**S_6**
S_1	void	yes	yes	yes	yes	yes
S_2	yes	void	yes	yes	yes	yes
S_3	no	yes	void	yes	yes	yes
S_4	no	yes	yes	void	yes	yes
S_5	no	yes	yes	yes	void	yes
S_6	yes	no	no	no	no	void

Table 3 contains transaction by command from one state to another for a collaborative system. Therefore, the change from $\boldsymbol{S_i}$ state to $\boldsymbol{S_j}$ state is done by $\boldsymbol{C_{ij}}$ command:

Table 3. Transaction by commands between collaborative systems states [2]

	S_1	**S_2**	**…**	**S_j**	**…**	**S_n**
S_1						
S_2						
…						
S_i				**C_{ij}**		
…						
S_n						

The transition from one state to another implies output delivery from the system. In a usual collaborative system changing from $\boldsymbol{S_i}$ state to $\boldsymbol{S_j}$ state is not possible for any ***i*** and ***j*** in ***1..n***. The system changes from one state to the other but it doesn't have the capacity to go from each state in all the others. The situation is met for instance when a very simple collaborative system like a virtual mono-product store markets cement. The possible states of the system are: open, closed, supplying, sale. Table 4 contains the system transitions between the four states:

Table 4. Transition between states for a virtual mono-product store [2]

	Open	**Closed**	**Supply**	**Sale**
Open	**void**	**yes**	**yes**	**yes**
Closed	**no**	**void**	**yes**	**no**
Supply	**yes**	**no**	**void**	**yes**
Sale	**no**	**yes**	**yes**	**void**

The change from *open* state to *closed* is done through **close store** command, while the transition from *supply* to *open* is triggered by the **open store** command. Getting from *open* or *supply* states to *sales* is done through **merchandise** command.

Probabilities are assigned to such a collaborative system, like: probability to change from $\boldsymbol{S_i}$ state to $\boldsymbol{S_j}$, probability to deliver ***x*** as output when changing from one state to another. These probabilities are determined by a series of parameters like: amount of stock-in-trades, the deposit capacity of the store, working program, customers' portfolio or the amount of daily orders. The systems probability to transit from *open* state to *closed* is influenced by the following situations:

- is the end of the working program;
- the stock-in-trades is used-up, following change to *closed* state and then to *supply*;
- the clients' orders are not honored because of different reasons.

Probabilities of these transitions are used in determining the appearance frequencies of different states of the systems at a given moment. According to Table 4, the change from the state of *supply* to the same state is not possible. In certain situations – like a large order from one of the clients, the system will go from one supply to other in order to honor the customer's demand. But the probability of such situations is low, considering that one supply loads the whole deposit capacity of the store.

Let Px_{ij} be the probability that the collaborative system representing a virtual mono-product store provides output x when changing from S_i state to S_j state. Px_{ij} is determined by the relation:

$$Px_{ij} = \frac{CFx}{CPx},$$

where:

Px_{ij} – probability that the collaborative system provides output x when changing from S_i state to S_j state;

CFx – number of favorable cases to obtain output x when changing from S_i state to S_j state;

CPx – number of possible cases to obtain output x when changing from S_i state to S_j state.

The *bank* collaborative system contains identities which generate messages like: demands for problem solution, open accounts, currency discharge, credit approval.

The identities are professional and in the job description they make certain operations. The person P_i operates n_i transactions: $O_{i1}, O_{i2}, \ldots, O_{ini}$

Each transaction implies certain documents: $d_{i1}, d_{i2}, \ldots, d_{ini}$

Each transaction has a solution: $s_{i1}, s_{i2}, \ldots, s_{ini}$.

All of these are stored in a very large database.

The dynamics of the collaborative systems point to modifications in the quality, structure, functions, dimensions, procedures and standards of the systems. The dynamics are studied by mathematical analysis, forecasting the long term behavior of each system [4].

2. The database structure

In [5] a collaborative system is presented for very large datasets visualization using web services. Web services implement collaboration and visualization through internet of very large data sets.

An article from the very large database contains: name of the person, the operation, the time of demand reception, the time of solution delivery (the time between the two times is the solving period) and the solution given.

```
struct article
{
char *person;
char *demand;
int input_moment;
int output_moment;
char *product;
};
```

Person: *Johnson*
Demand: *Taxi-cab authorization*
Input moment: *10*
Output moment: *11*
Product: *Yes*

This data is recorded by:

- the solicitor who tells his demand; the moment of demand reception is stored along with the address where the solution has to arrive;
- the solution giver;
- the auditor who verifies the quality of system activity.

The system activity leads to generation of *C* collection with *N* elements, *N* very large ($N>10^5$). The elements e_1, e_2, ..., e_N have identical structure and conform with the unique description template. It contains the list of the field-attributes of each article, the order of appearance and the type of each attribute. The elements are updatable and the rate *r* of field update is known.

Let *L* be the number of locations where the *N* elements are placed and let l_1, l_2, ..., l_i, ...,l_L be the number of elements belonging to the *i* location, so that $\sum_{i=1}^{L} l_i = N$. The local collectivities are formed through determined aggregation criteria: geographical, structural, rank, etc. The local collectivities c_1, c_2, ..., c_L are distinct physical databases.

Because of the large number of elements of any local collectivity c_i, the copy cost of elements is noticeable high. Therefore, in order to create the initial *C* collectivity with *N* elements a virtual concatenation of the *L* distinct databases is chosen. The central database integrates local collection by their description, functioning like a database concordance. Each record has a physical address and its own update data. In the moment when an update occurs, the data about it is stored in the local database c_i by modifying a single file which contains data about all the records in the collection. The concatenated database is updated by the automatic or manual copy of the *L* content files, when they are altered. The content files of the local databases can be database concordances of other smeller database of inferior hierarchic level.

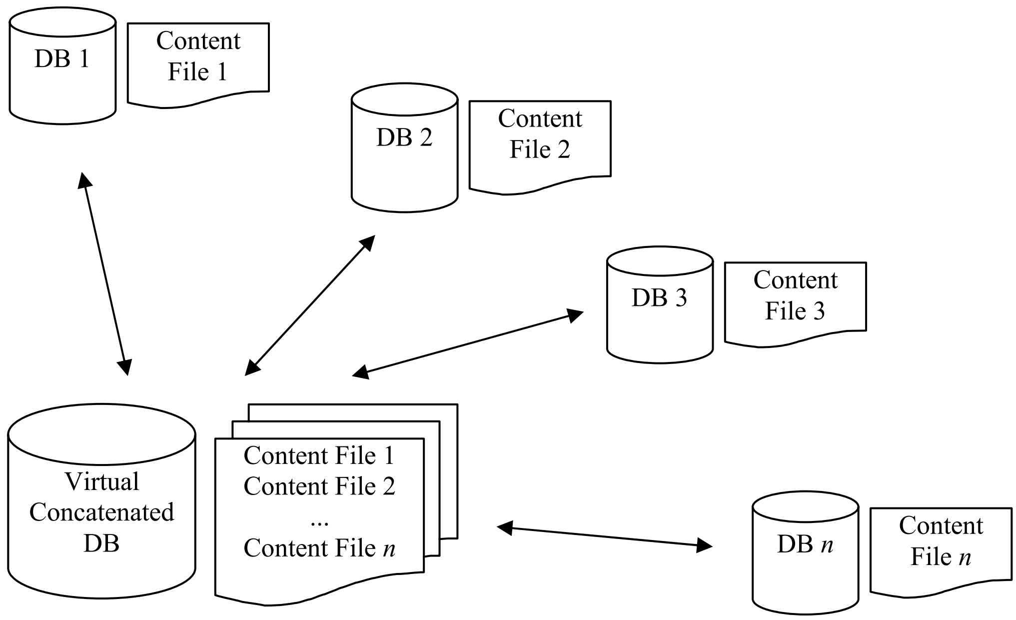

Figure 1. Virtual aggregation of DB

The daily data volume from the virtual database follows the relation:

$$VD_z = NP * \sum_{i=1}^{ND} NC_i$$, where:

NP – number of persons;
ND – number of documents;
NC_i – number of fields in the *i* document.

Data volume for a *k* days period is:
$VD_{k\text{-}days} = k * VD_z$, where:
VD_z – daily data volume.

Let *DB* be a database whereat *N* stations have access for creating and updating records. In the $[t_i; t_{i+1}]$ time period there are P_i persons that operate updates on the database, $i=1,\ldots,N$. The total number of persons that interacts with *DB* in the $[t_i; t_{i+1}]$ period is modeled by the relation:

$$NTP_j = \sum_{i=1}^{N} P_i$$

If each P_i makes k_i operations on *BD* and each k_i materializes in a new record in *DB*, then, the total number of records generated in the $[t_i; t_{i+1}]$ interval is:

$$NT\hat{I}_j = \sum_{i=1}^{N} P_i k_i$$

Therefore, for any two intervals $[t_i; t_{i+1}]$ and $[t_{i+1}; t_{i+2}]$ the number of records is exactly determined by the addition of the above indicators. Through gradual aggregation the long time generated data is determined.

This daily or periodic data volume includes an image of all that is realized in the collaborative systems in the informational field.

The records in the database are done:

- online by the solicitor;
- online through documents management by the agents of the collaborative system.

The documents' processing means extracting the information from the data fields and converting as strings of bits, which can be stored in a database. The processing of forms is considered completed when all the information from documents has been extracted and saved within a database [4].

In [6] a mechanism for auto-organization of devices collection is presented, for processing documents in a collaborative system from a multi-agent perspective.

The input data quality determines the quality of the information that is given to the users or to the decisional authorities within the system. The erroneous data has to be minimized from the initial phase. The validation procedures are grouped in:

- traditional procedures which consist in visual examination of data in primary documents;
- automated data control and validation procedures, using validation software and automated data correction procedures; the errors are automatically identified and corrected without any human intervention [4].

The forms contain required and optional fields. The input data is consisted of: letter strings, dates, codes or numerical values. This data frames in the following categories:

- correct and complete; a form contains the fill-in fields C_1, C_2, …, C_n; each field C_i has a value domain V_i; the *correct* and *complete* state is achieved when all the fields C_i, $i=1..n$, belong to vocabularies V_i, $i=1..n$;
- correct but incomplete; all the fields C_1, C_2, …,C_{i-1}, C_{i+1}, …, C_n belong to vocabularies V_i, $i=1..n$, but field C_i is missing; in this case, the application generates messages which indicate those fields which were not filled in;
- complete but incorrect; all the fields are filled in, but one or more values doesn't belong to the value-domain; in this case, the message generated specifies for each field what went wrong.

For each attempt of recording data, if the data is not completely correct and complete, a list of errors appear and the data is not sent to the database [7].

The application Collaborative Multicash Servicedesk records and processes phone demands taken by the analysts in a bank.

The fields that are selected or completed by the analyst are the following:

- name of the client which is selected from a predetermined list of Multicash users;
- name of the person which called;
- demand category, by selecting from a list with categories and associated codes;
- demand description for problem details;
- solving method by selecting the adequate option.

When a negative resolution is adopted to a credit demand, the justification is based on:

- incomplete documents in the credit;

- failing to fulfill the existing law requirements;
- failing to fulfill the technological restrictions: the wage is too small or assurance is inexistent.

Denials are coded for storing and for easy telling of the reasons of decision. Based on these codes reports and statistics are generated for classifying the denials on the type of negative responses.

3. Processing

Based on the very large datasets, many processing operations are realized in order to correlate and compute certain indicators.

Groups of persons are established and sorting takes place on the persons in the database leading to:

$Name_1$: t_1 articles;
$Name_2$: t_2 articles;
...
$Name_k$: t_k articles.

In the collaborative system there are *k* agents. From the Collaborative Multicash Servicedesk database certain data has been extracted about the number of demands, by clients recorded in November 2009, presented in Table 5:

Table 5. Number of demands by clients

Client	Number of demands
Client Name 1	200
Client Name 2	180
Client Name 3	220
Client Name 4	100
Client Name 5	300

Analyzing the number of demands by clients its determined the number of distinct clients that recorded demands in a specific period.

In the present, is seeking the increase of services quality offered by collaborative banking systems, by introducing intelligent agents to help increase performance of these systems.

The intelligent agent means an entity performing certain operations independently on behalf of a third party. The agents have a number of attributes, their main attribute is autonomy. For an agent to be called intelligent, its autonomous nature must be flexible, meaning that:

- perceive the working environment and the appropriate response to the changes occurred;
- decide to action also in situations of not environment amending;
- ability to interact with other agents or even with the human agent, both to achieve the designed goals and to facilitate the work of other employees.

Characteristics of an agent are the followings:

- mobility, which is defined as the ability to move in an electronic network;
- veracity, which implies that an agent is unable to provide false information;

- goodwill, which means that an agent does not have conflicting goals;
- rationality, which means that an agent acts to achieve the purpose.

An agent is characterized by an architecture and a program. The agent program is a function that matches the perceptions which the agent receives from the environment and its actions. This program must be compatible with the agent architecture. The architecture made the interface between the perception given by sensors and the program, run the program and ensure that all actions chosen are made as they are generated. The environment where an agent act has many facets, being fully or partially observable, deterministic or stochastic, static or dynamic, discrete or continuous, monoagent or multiagent.

The sort is done by messages or documents received and results:

Docum$_1$: k_1 *appearances;*

Docum$_2$: k_2 *appearances;*

...

Docum$_h$: k_h *appearances.*

It follows that in the system are *h* types of documents.

They are sorted by resolution and appear:

Yes: k_1 *appearances;*

Positive opinion: k_2 *appearances;*

...

Rejected: k_r *appearances.*

It follows that in the collaborative systems are *r* types of resolutions.

In the same manner are analyzed the evidence answers.

Incomplete documents: x_1 *appearances;*

Violation of legal provisions: x_2 *appearances;*

Technological incompatibilities: x_3 *appearances.*

Table 6 presents a report from the database of Collaborative Multicash Servicedesk application, regarding the categories of requests and their frequency in the month of November 2009:

Table 6. Number of requests by category

Category	Number of requests
Add new accounts in the client application	26
Add new users in the client application	14
Other requests	132
User blocked on the communication	41
User blocked at logon	20
Communication initiated	54
Confirm account balance	71
Deactivate payments file	1
Error on starting the application	5
Signature error	46
Error on see statements	20
Statements export	1
Generate electronic signature	20
Index corrupted in database tables	4

Category	Number of requests
Training on using the application	14
Training on see rejected payments	4
Delivery account statements	7
Delivery file with bank codes	12
Delivery files for distributed signature	8
Change communication channel	1
Change number of approvals / amount limits	1
Change name / address of payer	1
Move the application on another computer	13
Please repeat job with AC29	9
Reinstalling the application	7
Setting print parameters	5
Setting communication sessions	1
Training of branches for completing annexes	10
Transmission interrupted	36
Check payments status	162

On the basis of processing performed on the data sets, sorting is made by a single feature and are determined the metrics:

- average number of documents per person, *NDP*:

  $NDP = \frac{NTD}{NP}$, where:

  NTD – total number of documents in the system;
  NP – total number of persons.

- average number of refusals to 100 requests, *NR*:

  $NR = \frac{NTR}{NS} * 100$, where:

  NTR – total number of refusals in the system;
  NS – total number of requests.

Other statistics are performed in order to be used in the justice to determine the correlations between documents, customers, requests, and resolutions.

4. Combined analysis

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.

There are considered *M* large databases BD_1, BD_2,... BD_M, made with the same informatics application, reflecting data which characterize *M* collectivities ordered in disjoint territorial areas. It builds an informatics application for realize the virtual database *BDV* by an operation of concatenation of the basic data extracted from databases BD_1, BD_2,... BD_M.

The selection of records from the virtual database *BDV* requires to:

- across the data set of essential results in the concatenation process;
- identify the BD_i databases containing records associated to the selected essential data;
- take information from real databases for processing selected records;
- carrying out processing.

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

Popescu: settle rents documents, settle taxi permits, resolve global tax.

Ionescu: settle construction permits.

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.

From the combined analysis Analyst – Category of requests, on the basis of records from the Collaborative Multicash Servicedesk application, results that the analyst *Mihai Iancu* 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 *Marian Neagu* 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 *Mihai Iancu* has been overloaded.

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

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

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

If $x > z$, then Popescu gave more positive resolution than Ionescu. If $x > y$, then Popescu gave more positive than negative resolution. If $z > w$, then Ionescu gave more positive than negative resolution.

By generalization, being considered the data sets $D_1, D_2, \ldots, D_n$, 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:

- 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} – the total number of resolutions adopted by D_i;

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

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

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

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

From the calculation of the quantitative and qualitative indicators result the current status of collaborative systems and the elements requiring replacement or improvement.

5. Construction of procedures for online problem solving

The document is followed from entry until it will be solved in order to see what is doing. There are identified the routines and activities that are repetitive. The objective of online problem solving procedures is to automate the repetitive activities in order to increase the response time and to resolve the requests.

The collaborative systems have: structure, purpose, flows, resources, routines.

For the real organization that works is built its entry into the database, from which are taken rules, using large volumes of data on which are made selections, sorting, regrouping, extractions.

From reality is issued the collaborative system model, identifying:

- the number of real states;
- the real list of states;
- types of resources;
- the real list of resources;
- number of activities types;
- list of activities;
- durations or differences between final and initial moments.

It follows the real image of the system, from which are obtained the basic parameters of the model, average durations, limitations of resources, on which the model is built.

The collaborative system has associated a neural network giving automatic solutions.

The neuronal network deliver output data and the input data are taken from databases.

Having start and final moments available, the durations are determined and, by their size, frequencies are calculated.

There are determined the probabilities that durations will be equal to a given value, *P(duration = X) = Y*, or the durations to be less that a given value, *P(duration < Z) = W*.

A customer arrives with a request, enter to the portal, is called the neuronal network, resulting the proposed solution which is provided to the customer and he decides. This workflow is shown in Figure 2:

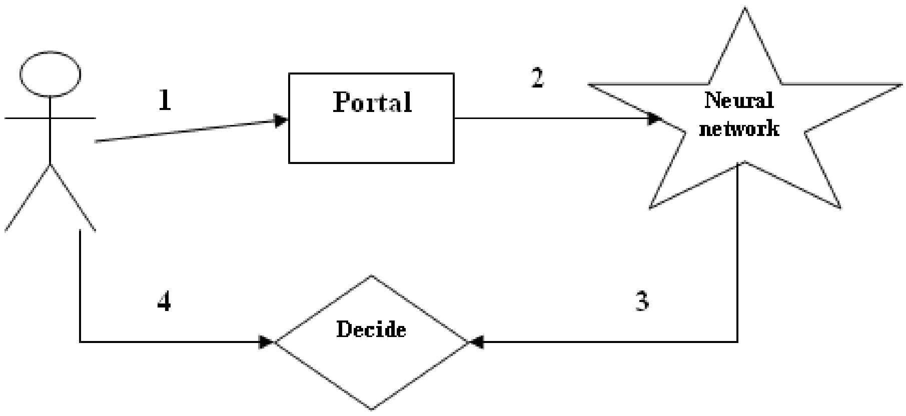

Figure 2. Workflow in a collaborative system

In a collaborative banking system, the customer interaction with the bank is done in several ways, some of which requiring a minimal effort from the client. On the banks websites are a series of applications and simulators to calculate interest rates on loans or deposit. Figure 3 shows an example of simulator for the calculation of receivable interest related to a term deposit:

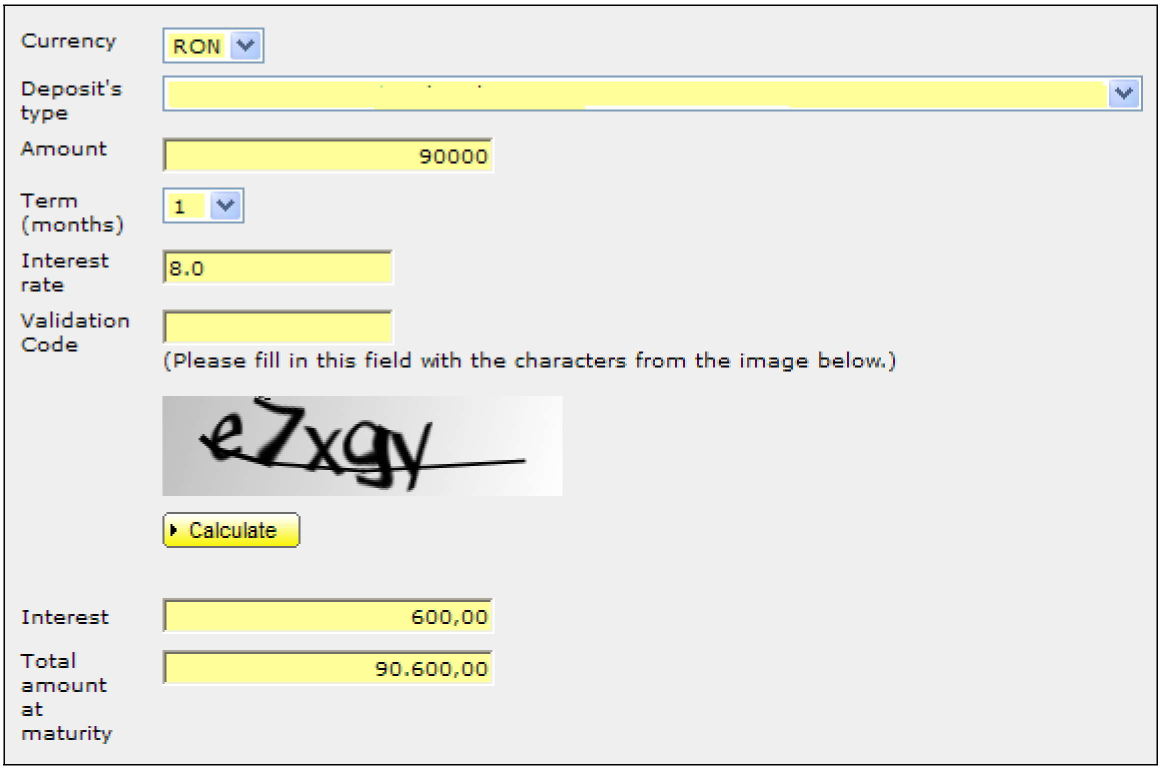

Figure 3. Deposits calculator (http://www.raiffeisen.ro)

Another example of simulator is an application for credit related calculation. The customer enters the bank website, select the type of loan, grant period, currency and interest type and enter the amount required.

The application will display to the user the monthly rate and amount of fees. These calculations are carried out also in the bank agencies, but this involves the customer go to the bank.

6. Conclusions

Collaboration is better appropriated if is based on simple rules, leaving the agents to fulfill their interests within their societies.

Increasing the volume of information and improving the software products for exploit it have led to a new quality of data usage by analysis that reveal to the organization's management information difficult or impossible to obtain otherwise. In this way are obtained information on customer preferences, their profile or distribution.

A collaborative system is characterized by a diversity of states, its transition from one state to another being accomplished through a document or command. The set of collaborative system states is finite, with an initial state and a final one. The system throughput between the initial state, the intermediate states and the final state, carrying out a cycle when it complete the transition from the final state to the initial state.

References

1. Dobrican, O. **An Example of Collaborative System**, International Workshop Collaborative Support Systems in Business and Education, Risoprint, Cluj-Napoca, October 2005, pp. 48
2. Ivan, I., Boja, C. and Ciurea, C. **Collaborative Systems Metrics**, Bucharest: ASE Publishing House, 2007
3. Ivan, I. and Ciurea, C. **Quality Characteristics of Collaborative Systems**, International Conference on Advances in Computer-Human Interaction, ACHI 2009, pp. 164-168, 2009, Second International Conferences on Advances in Computer-Human Interactions, 2009
4. Ivan, I. and Ciurea, C. **Using Very Large Volume Data Sets for Collaborative Systems Study**, Informatica Economica Journal, Vol. 13, No. 1, 2009
5. Zhao, Y., Hu, C., Huang, Y. and Ma, D. **Collaborative Visualization of Large Scale Datasets Using Web Services**, Internet and Web Applications and Services, 2007, ICIW '07, Second International Conference on, pp.62-62, 13-19 May 2007
6. Gnanasambandam, N., Sharma, N., Kumara, S. R. and Hua, L. **Collaborative Self-Organization by Devices Providing Document Services - A Multi-Agent Perspective**, Autonomic Computing, 2006, ICAC '06, IEEE International Conference on, pp. 305-308, 13-16 June 2006
7. Ivan, I., Ciurea, C. and Milodin, D. **Validarea datelor de intrare în aplicaţiile informatice orientate spre cetatean**, Revista Romana de Informatica si Automatica, Vol. 18, Nr. 4, 2008

---

[1] **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.

[2]Ion IVAN has graduated the Faculty of Economic Computation and Economic Cybernetics in 1970. He holds a PhD diploma in Economics from 1978 and he had gone through all didactic positions since 1970 when he joined the staff of the Bucharest Academy of Economic Studies, teaching assistant in 1970, senior lecturer in 1978, assistant professor in 1991 and full professor in 1993.

Currently he is full Professor of Economic Informatics within the Department of Computer Science in Economics at Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies. He is the author of more than 25 books and over 75 journal articles in the field of software quality management, software metrics and informatics audit.

His work focuses on the analysis of quality of software applications. He is currently studying software quality management and audit, project management of IT&C projects. He received numerous diplomas for his research activity achievements.

For his entire activity, the National University Research Council granted him in 2005 with the national diploma, Opera Omnia. He has received multiple grants for research, documentation and exchange of experience, conferences and congresses at numerous universities from Greece, Ireland, Germany, France, Italy, Sweden, Norway, United States, Holland, Australia, China and Japan.

He is distinguished member of the scientific board for the magazines and journals like:

- Economic Informatics
- Economic Computation and Economic Cybernetics Studies and Research
- Romanian Journal of Statistics

He has participated in the scientific committee of more than 20 Conferences on Informatics and he has coordinated the appearance of 3 proceedings volumes for International Conferences. From 1994 he is PhD coordinator in the field of Economic Informatics.

He has coordinated as a director more than 15 research projects that have been financed from national and international research programs. He was member in a TEMPUS project as local coordinator and also as contractor in an EPROM project. Also, he was expert assessor in many research programmes like RELANSIN, INFOSOC, CALIST and CEEX. He holds many awards and diplomas in research activity.

His main interest fields are: software metrics, optimization of informatics applications, developments and assessment of the text entities, efficiency implementation analysis of the ethical codes in informatics field, software quality management, data quality management and so forth.

List of Main Publications (2006 – 2009)

Ion IVAN, Marius POPA and Paul POCATILU (coordinators) – **Structuri de date,** ASE Printing House, Bucharest, 2008, vol. I Tipologii de structuri de date; vol. II Managementul structurilor de date

Ion IVAN and Catalin BOJA **Practica optimizarii aplicatiilor informatice,** ASE Printing House, Bucharest, 2007

Ion IVAN, Catalin BOJA and Cristian CIUREA **Metricile sistemelor colaborative,**: ASE Printing House, Bucharest, 2007

Ion IVAN, Gheorghe NOSCA, Sergiu CAPISIZU and Marius POPA **Managementul calitatii aplicatiilor informatice,** Bucharest: ASE Printing House, 2006

Ion IVAN, Traian BADICA and Marius POPA **Procese de agregare software,** Studii si Cercetari de Calcul Economic si Cibernetica Economica, vol. 42, no. 1, 2008, pp. 69 – 85

Ion IVAN, Eugen DUMITRASCU and Marius POPA **Evaluating the Effects of the Optimization on the Quality of Distributed Applications,** Economic Computation and Economic Cybernetics Studies and Research, vol. 40, no. 3-4, 2006, pp. 73 – 85

Ion IVAN, Cătălin BOJA, Marius VOCHIN, Iulian NITESCU, Cristian TOMA and Marius POPA **Using Genetic Algorithms in Software Optimization,** Proceedings of the 6th WSEAS International Conference on Telecommunications and Informatics, Dallas, Texas, USA, March 22 – 24, 2007, pp. 36 – 41

Ion IVAN, Marius POPA, Catalin BOJA, Cristian TOMA and Dragos ANASTASIU **Software for Structured Text Entities Dependency Graph Building,** Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications, Gold Coast, Queensland, Australia, January 17 – 19, 2007, pp. 224 – 230

Ion IVAN, Cristian TOMA, Marius POPA and Catalin BOJA **Secure Architecture for the Digital Rights Management of the M-Content,** Proceedings of the 5th WSEAS International Conference on Information Security and Privacy, Venice, Italy, November 20 – 22, 2006, pp. 196 – 201

[3] Cristian CIUREA has a background in computer science and is interested in collaborative systems related issues. He has graduated the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2007. He is currently conducting doctoral research in Economic Informatics at the Academy of Economic Studies. Other fields of interest include software metrics, data structures, object oriented programming in C++ and windows applications programming in C#.

[4] Sorin PAVEL has graduated the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2008. He is currently following Master's in Software Project Management and the Doctoral School in Economic Informatics, both at the Academy of Economic Studies.

[5] Codification of references:

[1]	Dobrican, O. **An Example of Collaborative System**, International Workshop Collaborative Support Systems in Business and Education, Risoprint, Cluj-Napoca, October 2005, pp. 48
[2]	Ivan, I., Boja, C. and Ciurea, C. **Collaborative Systems Metrics**, Bucharest: ASE Publishing House, 2007
[3]	Ivan, I. and Ciurea, C. **Quality Characteristics of Collaborative Systems**, International Conference on Advances in Computer-Human Interaction, ACHI 2009, pp. 164-168, 2009, Second International Conferences on Advances in Computer-Human Interactions, 2009
[4]	Ivan, I. and Ciurea, C. **Using Very Large Volume Data Sets for Collaborative Systems Study**, Informatica Economica Journal, Vol. 13, No. 1, 2009
[5]	Zhao, Y., Hu, C., Huang, Y. and Ma, D. **Collaborative Visualization of Large Scale Datasets Using Web Services**, Internet and Web Applications and Services, 2007, ICIW '07, Second International Conference on, pp.62-62, 13-19 May 2007
[6]	Gnanasambandam, N., Sharma, N., Kumara, S. R. and Hua, L. **Collaborative Self-Organization by Devices Providing Document Services - A Multi-Agent Perspective**, Autonomic Computing, 2006, ICAC '06, IEEE International Conference on, pp. 305-308, 13-16 June 2006
[7]	Ivan, I., Ciurea, C. and Milodin, D. **Validarea datelor de intrare în aplicațiile informatice orientate spre cetatean**, Revista Romana de Informatica si Automatica, Vol. 18, Nr. 4, 2008