Collaborative systems metrics

ION IVAN, PhD
ACADEMY OF ECONOMIC STUDIES BUCHAREST
Email: ionivan@ase.ro

Catalin Boja
ACADEMY OF ECONOMIC STUDIES BUCHAREST
Email: catalin.boja@ie.ase.ro

Robert Eniedy
ACADEMY OF ECONOMIC STUDIES BUCHAREST
Email: renyedi@yahoo.com

Popa Marius, PhD
ACADEMY OF ECONOMIC STUDIES BUCHAREST
Email: marius.popa@ase.ro

Cristian Ciurea
ACADEMY OF ECONOMIC STUDIES BUCHAREST
Email: cristian.ciurea@brom.ro

Cristian Toma
ACADEMY OF ECONOMIC STUDIES BUCHAREST
Email: cristian.toma@ie.ase.ro

Abstract. The paper describes the key elements of the collaborative systems. There are presented main characteristics for the collaborative systems. The paper analyzes different types of indicators. They represent the base for further metrics definition. There are described the indicators most important characteristics as sensitivity, non-

catastrophic, non-compensatory and representativity. The paper takes into discussion different applications for collaborative systems that describe particular metrics.

1. COLLABORATIVE SYSTEMS

Information and collaborative technologies are the key elements for any modern organization. The intranet offers the hardware support for computerizing the activities of the organization. There are three distinct goals for using the intranets [1], figure 1.1: information, communication and collaboration.

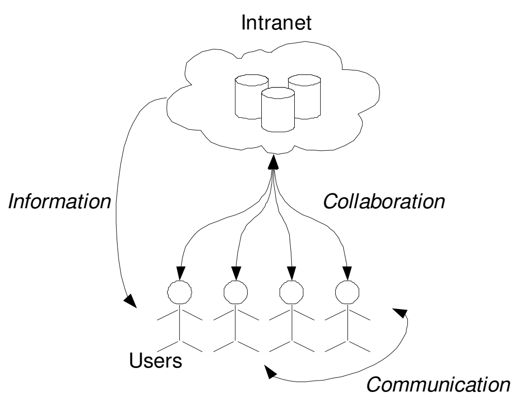

Figure 1.1. **The uses of the intranet inside an organization.**

Information is organized and presented in the organization's intranet as document management software, file sharing, employee files, internal news and various other kind of information. In the *information* process the users act as information receivers.

Communication in the intranet is done by the means of e-mail, mailing lists, content publishing tools, shared agendas, task management, instant massaging, e-learning. For content publishing the Wiki technology has become widely used. The software tools based communication involves the users as both information emitters and receivers.

Intranet based *collaboration* happens on discussion forums, activity created communities, information creation communication tools. The collaboration activity is a complex mechanism which engages users as information agents, receivers and emitters. The role of agent confers individuality to each user. Existing information is revised and gradually improved by the interaction of users. A software designed specifically for collaboration purposes reduces the volume of e-mails and improves the quality of information.

Intranets combine these goals so that the division between them becomes blurred. Collaborative intranets create a virtual marketplace of information and collaboration where each participant establishes what existing information he or she needs and what new information he or she will provide to the virtual community. The communication mechanisms are used for user notifications on the evolution of the quantity of the information available in the network. Each user becomes responsible for publishing his own content. The value of the contributions is determined by its popularity index.

No user from the organization should be limited in the interaction options with other organization members or members of the extended organization. Any limitation on the interaction attempts due to reasons of organizational hierarchy, confidentiality or tradition will most probably produce more negative than positive results. An intelligent organization automatically adjusts its information so that each user becomes responsible for the used and contributed information. Involving clients and business partners in the process of collaboration in an organization produces potential cost reductions [2].

At the same time the accumulated information in the collaboration process needs to be protected against destruction and theft. The most important defensive activity is the protection of the integrity of the information storage devices because a negative event of this nature puts in danger the whole information foundation of the organization.

There is a tight link between collaborative systems and knowledge management. The present implementation of knowledge management systems serves as:

- support for decision making by increasing the quantity and quantity of the available information,
- innovation by creating an optimal environment for identifying, validating and productizing of new ideas,
- optimization of production by developing best practices, reducing errors and reusing knowledge and know-how.

These items define a functional knowledge management which serves the various internal process of the organization. Collaborative intranets promote a new knowledge management type by the means of the operational knowledge management. In this variant, knowledge management facilitates and improves the quantity and quality of intellectual cooperation and human connections.

The concept of e-collaboration refers to computer assisted collaboration activities between users from disparate location. The users community in this case constitutes a virtual community and the activities are transposed in the perspective of the Internet, figure 2.1.

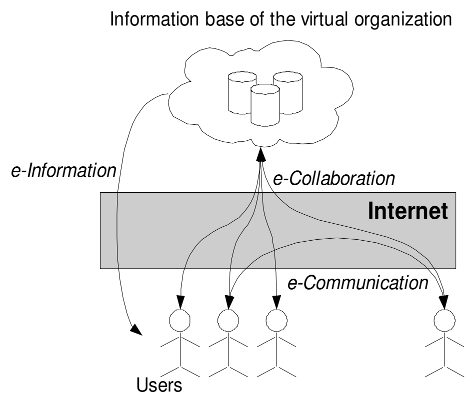

Figure 2.1: **The activities of a virtual organization.**

From this derives the concept of e-management which integrates information and collaboration technologies. There are a series of collaboration systems which lay the foundation of the e-management activity:

- enterprise resource planning – ERP – systems;
- e-procurement systems;
- efficient consumer response (ECR) technologies for managing relations between distributors and suppliers;
- customer relationship management (CRM);
- supply chain management (SCM);
- human resources information systems (HRIS);
- e-business systems;
- employee relationship management (ERM);
- groupware.

The central purpose of e-management is the intellectual cooperation. e-collaboration networks tend to have deficiencies at the level of team management due to the higher complexity of human relationships and information interchange. The reasons range from the technical level to cultural, organizational and methodological levels.

2. COLLABORATIVE SYSTEMS CHARACTERISTICS

The collaborative systems represent, from the implementation viewpoint, software entities that are developed during a life cycle process that starts with the problem analysis and ends with the implementation of a fully functional software 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. In the end, there are defined aggregate levels for common characteristics.

In order to describe the system overall quality level, there are defined models which take into consideration the importance of each characteristic.

Being a software product many of the characteristics are derived from the software analysis that is applied on interfaces, source code, modules and performance [9], [10].

From the viewpoint of the collection of components and links, the set of characteristics has a particular form.

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.

Based on the initial form of Halstead metric for sources code complexity it is defined a model to measure the complexity of collaborative systems. From the viewpoint of links and components framework , the complexity, K_T, takes into consideration the number of components, N_1, and the number of links, N_2, resulting:

$$K_T = N_1 \log_2 N_1 + N_2 \log_2 N_2$$

The complexity of the static graph, K_S, takes into consideration the graph associate to a collaborative system in which nodes, N_N, are the components and the arcs, N_A, are described by the message links between the components. It results the formula:

$$K_S = N_A - N_N + 2$$

As the initial form of Halstead metric, the relation highlights high levels of complexity for high determined values. From the viewpoint of the collaborative systems this describes a high level of interlinking between components and a good communication environment. Also, these imply a great care in managing all these connections and the redundancy of data being transmitted.

The *collaboration level* describes the number and types of links between system components. In order to allow communication between its components the system must contain links between its nodes. These are used to exchange data between different parts and to provide the communication infrastructure needed in a collaborative environment.

To measure the collaboration level there are defined indicators that measure and describe the degree in which one component is linked to the system. Considering the collaborative system as a tree structure, there are taking into consideration:

- the degree of vertical collaboration as the number of links between components from level k to the ones on level k+1;
- the degree of horizontal collaboration as the number of links between components on same level;
- the degree of total collaboration as the number of links between on component to every other components in the system; this indicator may be applied to any collaborative systems not taking into consideration their structure; its value is determined based on the relation:

$$TC = \frac{TNL}{TNC}$$

where:

TNL – total number of links;
TNC – total number of components.

If the indicator value, TC, is equal with the number of components from the system it means that every node is linked with all the others by a direct connection. That influences the speed of communication decreasing it but it also increase the system complexity. Otherwise, if the value is smaller than TNC-1, it means that some components are not connected with the system and that affect the collaborative aspect.

This characteristic may be analyzed also from the viewpoint of the length of the track the message is taking from the source component to the destination one. On this way, the system must take care the messages are not lost in the system or they aren't altered.

System *reliability* is a very important quality characteristic because:

- it value is directly determined by the number of processes and activities that give correct and complete results;
- allows particular approaches for determining models of quality estimation; taking into consideration the hypothesis that once the causes that generates unwanted errors and system failures are eliminated it is possible to increase its levels and directly the system quality;
- its value influences the entire collaborative system project;

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

$$I_{Fiab} = \frac{r_{succes}}{r_{total}}$$

where:

r_{succes} – the number of successful solved situations; this situations give complete and correct results;

r_{total} – total number of considered situations.

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.

Being a complex structure, the collaborative system must adapt to changes that occur in internal communication algorithms, entry data, rules, results. The changes to be implemented must be realized with minimum costs in time and financial resources.

Modifications on algorithms take place in modules where messages are evaluated or are defined components behavior. To develop maintainable systems it is necessary to define clearly modules and specifications. Vulnerable modules and components must be defined and developed in such a manner that they will not affect other parts of the system when they are modified.

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.

A detailed analysis of the system maintainability is achieved by taking into consideration characteristics like stability, analyzability, changeability and testability.

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

When referring to functionality the developers must take into consideration the implementation of security routines. Being a collaborative system the structure implements communication between its components. The importance of data and of decisions that are taken based on them has great impact on how the messages are protected against unauthorized access and corruption.

3. TYPES OF INDICATORS FOR COLLABORATIVE SYSTEMS CHARACTERISTICS ASSESSMENT

The analytical form of an indicator used to measured quantitative levels for collaborative systems qualitative characteristics is based on

$$y = f(x_1, x_2, \dots, x_{nfc}),$$

where:

- nfc – number of identified factors which have impact on the evolution of analyzed phenomena;
- x_i – measured level for the i^{th} influence factor of the case study;

f( ) – an analytical real form used to represent the dependency between the influence factors and result variables; it is used to describe and to study the phenomenon;

y – result variable that describe an existing situation in the phenomenon evolution.

The concrete analytical form of an indicator is:

- linear, if:

$$y = f(x),$$

$$y = f(kx)=kf(x),$$

and the graphic that describes the function values is represented by a straight line;
- non-linear, if the function graphic is irregular and it can't be represented by a straight line.

Linear form are described by functions which are:

- constant :

$$y = f(x) = a$$

with *a* real number constant;
- linear:

$$y = f(x) = a*x$$

with *a* representing a real number coefficient;
- generalized form for a linear function:

$$y = f(x) = ax + b$$

with *a* and *b* real number coefficients;
- composed from multiple functions:

$$\mathrm{y = f(x) = } \sum_{i=1}^{nfc} a_i x_i + \mathrm{b}$$

Nonlinear usual forms are represented by functions that are:

- power functions:

$$y = f(x) = ax^{nfc}$$

with *nfc* fixed natural number natural and *a* a real number constant;

- quadratic:

$$y = f(x) = \sum_{i=1}^{nfc} a_i x_i^2$$

representing particular forms for power function;

- polynomial:

$$y = f(x) = a_n x^{nfc} + a_{n-1} x^{nfc-1} + \dots + a_1 x + a$$

with $nfc \geq 2$ fixed positive integer and real coefficients;

- exponential:

$$y = f(x) = e^x$$

- logarithmic:

$$y = f(x) = log_b x$$

with positive base and $b \neq 1$;

- root:

$$y = f(x) = \sqrt[nfc]{x}$$

with *n* fixed natural number;

- fraction:

$$y = f(x) = \frac{g(X)}{h(x)}$$

where the *h(x)* function does not return zero values.

In order for values, that are associated with different levels of measured software characteristics, to be significant when they are pulled put from the model context, they must have measuring units. Examples of different measuring units used in the analysis of collaborative systems are given by the number of agents, operators, operands, structures, levels, communication mediums and channels. Considering collaborative systems the result of software development process then the collection of measuring units is completed by the number of instructions, code lines, fundamental data types, classes, errors, modified code lines and tests.

There are considered the factors $Factor_1$, $Factor_2$, …,$Factor_{nfactor}$ to which are associated the measurement units um_1, um_2, …,um_{num}. If it is considered the linear analytical form:

$$y = \sum_{i=1}^{nfactor} a_i x_i$$

where:

a_i – represents real number constants with positive values;
x_i – represents the measured level for the factor $Factor_i$;

then it results that the considered measurement units um_1, um_2, …,um_{num} are identical between them and the results also has the type *um*. For example if x_1, x_2, …,x_{nfc} are measured in meters then:

- the value for $y = \sum_{i=1}^{nfc} a_i x_i$, is also measured in meters;
- the value for $y = \sum_{i=1}^{nfc} a_i x_i^2$, is measured in square meters;
- the value for $y = \frac{x_i}{x_{i+1}}$,cu i = 1..*nfc*, represents an nondimensional number.

Any metric is the subject of a dimensional analysis in order to assure the aggregation correctness.

If

$$y = \frac{x_1 * x_2 * x_3}{x_4}$$

where x_1, x_2, x_3, x_4 represent four variables that have as measurement units:

um_1 – number of components;
um_2 – message medium size as number of bytes;
um_3 – number of messages;
um_4 – bandwidth size as bytes;

then the measurement unit for y, u_y, is:

$$um_y = \frac{um_1 * um_2 * um_3}{um_4} = \frac{[components\ \text{number}] * [bytes] * [messages\ \text{number}]}{[bytes]}$$

In the end the measurement unit is:

$$um_y = [components\ \text{number}] * [messages\ \text{number}]$$

Because um_1 is a nondimensional value, it results that

$$um_y = [\,lei\,]$$

Any measurement unit that is raised to a zero power it will represent the nondimensional measurement unit equal with *one*. Generally, the significance of a nondimensional unit is to represent the proportion from a total. In the case of nelinear models, the dimensional analysis is absolutely necessary.

Between the components of the international system for measurement units there are correlations, for example in the case of the physics laws:

$$[\text{Force}]_{um} = [\text{weight}]_{um} * [\text{acceleration}]_{um}$$

where $[\ ]_{um}$ represents the measurement unit of selected element;

There are considered the measurement units $um_1, um_2, \ldots, um_{num}$ and there are defined the operations:

$$um_i + um_i + \ldots + um_i = um_i$$

$$n * um_i = um_i$$

$$um_i * um_i * \ldots * um_i = \prod_{1}^{kum} um_i = (um_i)^{kum}$$

$$\frac{(um_i)^k}{(um_i)^r} = (um_i)^{k-r}$$

Also, in the case of dimensional analysis, conducted on software characteristics, it is necessary to define a systems of links between all the measurement units and to do a complete analysis.

4. INDICATORS PROPERTIES

When the software quality indicators are built, then the following elements are identified:

- the factor set that influence the quality characteristic;
- the variables that are associated to establishing of the procedures for measuring making;
- the software lots used for measuring of the variables that influence the quality characteristics.

For a quality characteristic, a lot of estimation indicators are built depending of work hypothesis and data gathering capability necessary for computation making. The indicator has an analytical expression easier or more complex depending of influence factors, influence intensity and reused structured of indicators with the behavior already known.

Also, the indicators for quantification of characteristic levels for maintainability, reliability, portability, complexity has a variety of analytical expressions, from homogeneous expressions to reports of homogeneous expressions, leading to constructions in which logarithmic and exponential function appear.

The analytical forms of the indicators must be built such as the indicators simultaneously assure the following conditions. They must be:

- sensitive, that is at small variations of the influence factors the result variable has small variations; at big variations of the influence factors the result variable has big variations;
- non-compensatory, that is at different variation sets of the factors, small values of the result variable are not obtained;
- non-catastrophic, that is at small variations of the factors, big variations of the result variable have not to obtain;
- representative, it represents the quality to be accepted by users in analysis making assuring the significance of the results.

To highlight when an indicator is sensitive, non-catastrophic or non-compensatory, there are built data sets for analysis.

The small variations are obtained using arithmetical progression, The big variations are obtained using geometrical pregression.

The indicator K_S used to measure the program complexity is sensitive because for a variation from N_A to N_A', with $N_A' = N_A + \gamma$ and from N_N to N_N', with $N_N' = N_N + \Delta$ it results

$$K_S' = K_S + (\gamma - \Delta).$$

This indicator is compensatory for the all cases in which $\gamma = \Delta$. The indicator K_S is non-catastrophic because at very small variations of the factors, N_A and N_N, very big variations of the indicator are not obtained. Also, there are not values of the factors that to lead to incapacity to compute the complexity value, in the same way with indicators of report type I=A/B, when the value of the element from denominator tends to null value.

The indicator $K_T = N_1 \log_2 N_1 + N_2 \log_2 N_2$ is sensitive because the variations from N_1 to $N_1' = N_1 + \gamma$, respectively from N_2 to $N_2' = N_2 + \Delta$ determines:

$$K_T' = N_1' \log_2 N_1' + N_2' \log_2 N_2' = (N_1+\gamma) \log_2(N_1+\gamma) + (N_2+\Delta) \log_2(N_2+\Delta) =$$
$$N_1 * \log_2(N_1+\gamma) + N_2 * \log_2(N_2+\Delta) + \gamma * \log_2(N_1+\gamma) + \Delta * \log_2(N_2+\Delta) > K_T + \gamma * \log_2(N_1+\gamma) + \Delta * \log_2(N_2+\Delta)$$

Because the operators have using precise rules, for indicator K_T some restrictions appear regarding the compensatory character. The variation domains for variables N_1 and N_2 are dependent of programming logic elements. There are not expressions in which operators without operands are used and expressions in which operands are defined without be used.

It considers the indicator GI that measures the objective Ob_1, Ob_2, ..., Ob_{no} accomplishment degree within analysis process of a collaborative system SC. The indicator is also used to analysis the system quality level, SC, through planned level accomplishment. The objective are depicted both quantitatively, and qualitatively, the two levels being measured through metrics. The analytical form of the indicator is:

$$GI = 0,6 * \sum_{i=1}^{n} pcalit_i \frac{MIN\{x_i, y_i\}}{MAX\{x_i, y_i\}} + 0,4 * \sum_{i=1}^{n} pcant_i \frac{MIN\{u_i, w_i\}}{MAX\{u_i, w_i\}}$$

in which:

- $pcalit_i$ – importance coefficient associated to the quality level for collaborative system, SC;
- x_i – planned level of the objective Ob_i quality;
- y_i – accomplished level of the objective Ob_i quality;
- $pcant_i$ – importance coefficient associated to quantitative level of the objective Ob_i from the whole quantitative level of the collaborative system SC;
- u_i – planned quantitative level of the objective Ob_i;
- w_i – accomplished quantitative level of the Ob_i.

The importance coefficients associated to the qualitative and quantitative level of the objective Ob_i are determined using methods depicted in [IVAN04a] such as to describe the quantitative or qualitative level structure in collaborative system ensemble. Considering that the objective set Ob_i, with $i = 1, \ldots, no$, describes in totality the collaborative system, then the importance coefficient levels describe the following expressions:

$$\sum_{i=1}^{no} pcalit_i = 1 \text{ and } \sum_{i=1}^{no} pcant_i = 1$$

The indicator GI values are included in the interval [0; 1]. In case in which no qualitative or quantitative planned levels are accomplished, having the value zero, for indicator GI the null value is obtained. This aspect indicates the plan accomplishment in zero proportion. If the planned levels are accomplished, then the indicator GI takes the value 1, highlighting a collaborative system obtaining with characteristics that are 100% conform to the specifications.

On the base of previous analysis, it established that the importance level of the objective quality represents 60% of general quality of the system, the rest being depicted by quantitative level of the objectives. From this reason, the partial sums associated to the objectives are weighted with 0,60, respectively 0,40.

The indicator GI is sensitive because any modification of the accomplished qualitative level y_i of the objective Ob_i, with the value $\alpha_{calit} = |y_i - y_i'|$, where y_i' represents the new value, leads to:

$$GI' = \begin{cases} GI - 0{,}6 * pcalit_i * \alpha_{calit}; \text{if } y_i' < y_i \text{ and } y_i' < x_i \\ GI + 0{,}6 * pcalit_i * \alpha_{calit}; \text{if } y_i' > y_i \text{ and } y_i' < x_i \end{cases}$$

what highlight a variation of $|0{,}6 * pcalit_i * \alpha_{calit}|$. If it also considers the quantitative variation of the objective Ob_i with $\beta_{cant} = |w_i - w_i'|$, where w_i' represents the new value, then the indicator has the variation:

$$GI' = \begin{cases} GI - 0{,}6 * pcalit_i * \alpha_{calit} - 0{,}4 * pcant_i * \beta_{cant}; \\ \qquad \text{if } y_i' < y_i,\ y_i' < x_i,\ w_i' < w_i \text{ and } w_i' < u_i \\ GI + 0{,}6 * pcalit_i * \alpha_{calit} + 0{,}4 * pcant_i * \beta_{cant}; \\ \qquad \text{if } y_i' > y_i,\ y_i' < x_i,\ w_i' > w_i \text{ and } w_i' < u_i \end{cases}$$

The indicator is compensatory in case in which the qualitative and quantitative levels associated to the objectives have a variation such as it obtains the same general level for analyzed collaborative system. For instance, it considers the objective Ob_i, that has a modification of accomplished quality level from the value y_i to y_i', with $\alpha_{calit} = y_i' - y_i$ and $\alpha_{calit} < 0$, and accomplished quantitative level increasing from w_i to w_i', with $\beta_{cant} = |w_i - w_i'|$. If the levels for the other objectives are not modified, then the compensatory character is highlighted for

$$0{,}6 * pcalit_i * \alpha_{calit} = 0{,}4 * pcant_i * \beta_{cant}$$

The indicator GI determination is based on design and planning process for the collaborative system. This is controlled by elements strictly linked of producer's resources and objectives. From this reason, the software analysis practice avoided the situations in which it considers the planned levels x_i and u_i equal with the value because these are the direct result of non-inclusion of the objective Ob_i in analysis process. Also, the cases in which the accomplished levels are much bigger than the planned ones are the results of a bad management that it must not appear in the good done projects. On the base of this hypothesis the non-catastrophic character of the indicator GI is highlighted.

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

$$R = \frac{K_{dec}}{Total_{dec}}$$

where:

K_{dec} – correct decision number based on indicator value;

$Total_{dec}$ –decision total number taken in analysis process of the collaborative system on the base of the indicator taken into account.

The 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 this situation, it concludes that analyzed metric is used for software analysis, representing a viable instrument to accomplish the objectives.

This approaching of the representativeness character analysis for software metric is an empirical one because it is based on experimental results, decisional process analysis after software analysis development. The disadvantage of the solution is given by the moment of determination for representativeness degree. In case in which the used metric has not the non-representativeness characteristic, the indicator R has the value R < 70%. This aspect implies allocation and inefficient using of resources in software analysis, it decreases the significance degree of the results and for the whole process, as result. The advantage is given by empirical character of the analysis.

This fact represents a strong argument to sustain either the significant or not significant character of the metrics.

Using the results and data of previous software analysis it is studied the significance of implemented metrics for well defined types of issues and collaborative systems.

Using this information, obtaining the R indicator for considered metrics represents one stage of analyzing the results.

In case of actual using the activities of software analyze it is difficult to determine the significant level of used metrics, based on R indicator. The reason is based on the unknown value of total number of decisions made based on the values of analyzed metrics.

The significance character is determined in this case by implementing statistic methods to analyze connection between data series and the significance of the factors.

Used indicators and statistic tests are:

- the report
- the F test
- the Mann-Whitney test
- the test χ^2

Unlike the other three properties, sensitivity, compensatory and noncatastrophic, the significance has a relative character, being dependent of the software analysis in which that indicator is used. Therefore, this indicator is or is not characterized by the considered property accordingly to the analysis's objective.

Testing the properties is done also using statistic methods that offer the necessary instruments to analyze the values sets. In order to emphasize the sensitivity of an indicator, the next algorithm is being used:

Step1: are created the terms of arithmetic progression related to the variation of influence factors

Step2: indicator value is calculated using the related software metric

Step3: the correlation coefficient is calculated for the result variable and each of the factors of metric's model.

If the correlation coefficient has a very high value it means that the indicators contains the property that is being analyzed.

For the metric K_T, associated to the complexity of a collaborative system, are considered the table 1 data.

Table 1. **Values for factors and for K_T metric.**

No.	**N_1**	**N_2**	**$K_T = N_1 \log_2 N_1 + N_2 \log_2 N_2$**
1	100	115	1451.62
2	110	120	1574.78

3	120	125	1699.55
4	130	130	1825.82
5	140	135	1953.47
6	150	140	2082.42
7	160	145	2212.6
8	170	150	2343.92
9	180	155	2476.33
10	190	160	2609.78
11	200	165	2744.21
12	210	170	2879.59
13	220	175	3015.86
14	230	180	3153
15	240	185	3290.96
16	250	190	3429.72
17	260	195	3569.25
18	270	196	3673.22
19	280	200	3804.97
20	290	205	3946.47

It is calculated the correlation coefficient between N_1 and K_T, equal to $r_{K_T/N_1} = 0{,}99$ and the correlation coefficient between N_2 and K_T, equal to $r_{K_T/N_2} = 0{,}99$. It is calculated an medium correlation indicator $\bar{r}_{K_T}$, whose relation is:

$$\bar{r}_{K_T} = \sqrt{r_{K_T/N_1} * r_{K_T/N_2}}$$

It is verified the significance of correlation coefficients by comparing the values with the 0.75 level. If the coefficients are higher than this value, it means that between these two variables is a strong connection, meaning that the variation of factors determines a variation within resulting variable.

In order to emphasize the compensatory character, the values sets are analyzed to point out the situations where the factors' variation leads to obtained values that are associated to the software metric and that are either constant or characterized by a very small variation. The determination of the values sets, that are required to test the non compensatory character, influences the verification process because it must be find out those factors values that, even if are different, describe a constant level or a lever with little variations for software metrics.

At the end, results a table where the factors' values varies but the same value is obtained in the indicator column. Finding the correlation coefficient in this case leads to values under 0,2, meaning that, between dependent variable and factors there is no significant relation. This situation is particular due to special nature of data and it mustn't be generalized on the entire metrics-factors relations.

For analyzing the catastrophic character must be taken into consideration sets of data that have descending values and that determine high values for the resulting variable. Also, there are taken into consideration the situations in which the ascending values determine low values for the resulting variable.

What stands out in this value variation is based on the modification rate of levels values, finally leading to the incapacity of find them out. In case of the type I indicators, that have analytical forms as

$$I = \frac{A}{B}$$

the catastrophic character is the result of the very high variation of metrics' value while the value of B factor is converging to zero.

For the K_T indicator the catastrophic character is emphasized by the values of N_1 and N_2 factors that are converging to zero. In terms of collaborative systems, this particular situation that indicates the lack of components and links, is inexistent in reality.

So, complexity metrics, like K_T, have a non catastrophic character.

5. APPLICATION OF COLLABORATIVE SYSTEMS

If ten different people are being interviewed regarding what means for them collaboration in informatics area there are received ten different answers. Some of them understand by collaboration changing e-mails. For others collaboration means videoconference through Internet. Most of people have difficulties defining this notion because in informatics have been implemented so many technologies and as a result the definition of collaboration is quite wide. Collaboration represents the integration of different technologies in one application that facilitates information share and management.

Integrated technology is only one aspect of collaboration as it is defined. The other aspect is timing. People are accustomed to real time collaboration, meaning working at the same time with other persons. New technologies offer an entire different way of collaboration, for example asynchronous collaboration, meaning that one doesn't have to be present in order to participate. Asynchronous collaboration allows us to collaborate with other people as we wish: through e-mail, Internet, Intranet and all the other asynchronous communication forms.

The design, development and use of collaborative systems have build a very interesting and complex case of crossed development between utility and software engineering. The design focused on users is a necessity for this kind of systems that allow different actors to work together in a cooperative environment. The classic interaction man – computer is enhanced by human-human interaction, HHI and by the need ok knowledge, [21].

Collaborative systems represent an interdisciplinary area situated at intersection of economy, informatics, management and sociology. Using IT technologies new collaboration opportunities were developed on the electronic products and services market. Collaboration involves organizations with same goals that are uniting in order to form a new structure. A collaboration example it is a strategic alliance.

A collaborative system represents the system where large numbers of users or agents are engaged in share activities, most of the times located in distant areas. Inside the large family of distributed application, the collaborative systems are distinguished by the fact that agents work together in order to reach the same objective and there is an important need for interacting with each other [19].

Collaborative recommendation is a wide used technique of giving access to personalized information. A system of recommendations holds the users' profiles and creates the recommendations based on similarities between users. This type of system keeps tracks of its users preferences and uses them in order to offer new suggestions. The collaborative recommendation systems are used for a large scale of recommendations, [17].

An informatics collaborative system is similar to a distribution firm whose objective is to sell bigger and bigger quantities of its products. The difference between a collaborative system and a distributed one is based on the following characteristics:

- the systems' elements, described both by users and agents, interacts with each other influencing the systems' behavior

- the systems' components use partial resources in order to fulfill both own objectives and common objectives.
- inside collaborative systems there are permanent channels of communication between users and agents.
- the agents have common and non opposing interests.

Cooperation and collaboration are synonyms in terms of acting or working together for reaching a common goal, even if in literature have different meanings. Roger and Johnson (2002) define cooperative learning as a connection between a group of students, requiring a positive dependence, individual responsibility, interpersonal abilities, interaction and analyses. Strijbos (2000) sees a distinction between cooperative learning and collaborative learning, based on the sum of predefined structures, the objective of studying and the group's size. These are developing a classification model in order to illustrate not just the differences between both perspectives but also the different types of support offered by he computer in group studying [20].

The systems contain the following elements:

- components
- components' interactions (exchange of messages and activities)

When collaborative systems are used in a voluntary way, one of the most important factors that leads to success is the manner in which the users feel their experience with the system: do they enjoyed it, does the system offer what is expected from it, are they capable of freely and natural communicate with other participants and do they want to recommended to other persons.

There are considered two components of a collaborative system, C_i and C_j, and two activities: A_k, activity made by C_i, component and A_{k+1}, activity made by C_j component.

From C_i to C_j is transmitted the M_1 message and from C_j to C_i is transmitted the M_2 message. This system is described in figure 5.1.

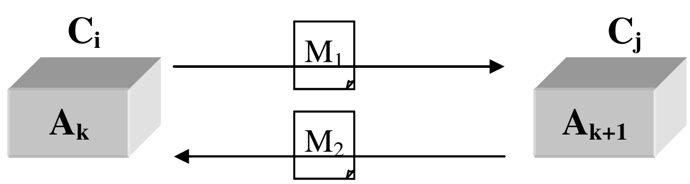

Figure 5.1. **Message transmission between system components.**

The M_1 message announces that the activity A_k has ended. This allows the execution of the A_{k+1} activity. The M_2 message is a confirmation from C_j that it has received the M_1 message.

In collaborative systems, the message exchange between C_i and C_j is limited. Between the two components there are transmitted $\boldsymbol{M_1, M_2, \ldots M_k}$ messages, with $k \geq 2$. From C_i component to C_j component are sent the $\boldsymbol{M_1, M_3, \ldots M_{2k+1}}$ messages. They have the property that $M_{2a+2} \cap M_{2b+1} \neq \Phi$

Objective definition in an activity is an iterative and convergent process. It is considered the collaborative system composed from two components I and J, representing two authors working on an article. The I author writes the T_1 text and the J author modifies it resulting the T_2 text. The system I component receives the T_2 text and modifies it into the T_3 text. The J component modifies the T_3 text and results the T_4 text. All these activities continues for a limited set of steps until the text represents on objective accepted by both of the two components

The modifications applied on initial text contains words from the V_I vocabulary and also from V_J vocabulary. The quality of the collaboration between the authors has a great level depending on the results of $V_I \cap V_J$ and comparing them with $V_I \cup V_J$. If the result of the intersection has K elements and the reunion has L elements then the collaboration base is a good one if the value of K is almost equal with L. The collaboration base is a limited one if K is much more smaller than L..

The collaboration process supposes mutual understanding between persons and a particular attitude.

An example of collaborative systems is the system composed from a group of authors which goal is to write together a paper in their domain. The collaboration in writing the article suppose:

- defining the group of authors based on common criteria: common background, same preoccupations, overall acceptance of a set of rules;
- the development of project structure from one of the authors, resulting the text T_1;
- common dialog over the text T_1 and the acceptance of the text T_2 by all authors; T_2 differs from T_1, but not in a significant way;
- task distribution between authors; the A author does the bibliography, B gives examples, C describes models, D writes software,

E describes entry data, F does the experiments and G puts together the work of all others;

Science has great impact on the development of different types of collaborative systems from various activity fields. One domain that was one of the first fields presenting great interest in implementing complex collaborative systems is the military. The complexity of exercises and military deployment of forces involves the coordination of many resources. The collaboration between different system components has a significant effect on achieving final objectives. Otherwise the results may conduct to a disaster.

Another example is given by the medical field in which modern communication technologies allow doctors from around the world to work on the same patient. In a chirurgical operation each person from the group of doctors has distinct roles.

In [22] it is analyzed a collaborative system model representing a training on different chirurgical activities that is done in a virtual medium. This example is based on the scenario in which the instructor and the trainee are on different locations. The instructor and the trainee share a common virtual space that contains various tridimensional anatomical models. Each person interacts with the other one through the virtual space and a medical simulation engine describes the physical and logical behavior of objects present on the virtual scene. The interaction is maintained by a multi-modal interface that uses visual 2D and 3D data, voices and audio simulation. Each person is in front of a working table that has a monitor and stereo active pair of glasses. All of these generate a tridimensional desktop. For collaborative use, it has been implemented a mini broadband system that allows creating a videoconference between persons.

The interaction between the instructor and the trainee is based on voice, gestures, chirurgical demonstrative actions, step by step tutorial and simultaneous actions.

The chirurgical training from this example, suppose a high level of interaction between the two persons. In opposition with training systems developed only for chirurgical dexterity, this process is concentrated also on procedures that target the understanding of the patient anatomy. The learning process is enhanced by the demonstration, the dialog and the show how activities.

Nathanael Thompson and Haiyun Luo from Urbana-Champaign University of Illinois have described an example of collaborative systems for Internet access in a residential area. The need for this project was given by the fact the clients Internet applications reach connection limits in the last kilometer of the area. In spite of the bandwidth of dial-up connections implemented in US residential areas through cable, the last kilometer of the infrastructure allow with great delays the use of shared files, downloads and web transfers. Because of that, many users have selected from a wide variety of broadband service providers. Some of them use cable connections DSL or satellite links.

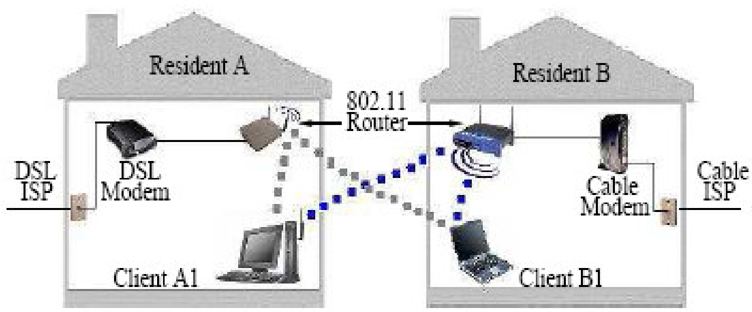

Figure 5.2. Local network for collaborative access to Internet.

This variety of links represents a great opportunity for neighbor users to share their Internet connections through high speed wireless networks. The objective is mutual benefit by improving the quality of the Internet connection. Figure 4 describes an example of shared connections between two neighbors, one using DSL access and the other one cable access. They plan the common traffic by implementing a wireless router.

6. CONCLUSION

The field of software metrics is a domain that has many published papers and that has acquired in the last period a great volume of theoretical knowledge. This provides the methods and techniques to analyze the problem, to identify the resulting variables, the influence factors and in the end to define the model.

The real problem is to apply the metric and most important to validate it. This will give the confidence that the values are real and the results are reflecting the actual image of the problem. Once the model is defined, it must be implemented in real development or maintenance cases and it must be tested.

The complexity of the collaborative systems on which there will be applied the metrics has great impact on the number of factors and as result on the scale of the model. In the end, it must be reached equilibrium between the model dimension and its capability to give significant results. The metric 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.

To define operational metrics for collaborative systems it is necessary to accomplish a series of stages:

- definition of dependent variables and of the exact way to measure them; throughout measuring tests there is highlighted the consistent character of the dependent variables set; all the elements that have an important role in the collaborative system architecture must e taken into consideration;
- development of auxiliary software applications that will measure in an automate manner the dependent variables levels;
- validation of measured values for determining if they are correct ;
- definition of exact rules for building test examples;
- assure the comparability of results by using same measuring procedures on predefined factors.

References

[1] Olivier Zara – Managing Collective Intelligence – Axiopole Publishing, 2004, http://www.axiopole.com/pdf/Managing_collective_intelligence.pdf

[2] Tony Kontzer – Collaboration System Cuts Unexpected Costs – InformationWeek, January 5, 2004, http://informationweek.com/story/showArticle.jhtml?articleID=17200033

[3] Ion Ivan, Felix Simion, Viorel Nica,- *Gradul de interferenta al metricilor Halstead si McCabe*, Revista Romana de Informatica si Automatica, vol. 7, nr. 3, 1997, pg. 23-28.
[4] Alexandru BALOG, Ion IVAN, Mihai POPESCU - *Extensii ale metricii Halstead pentru sisteme de programare*, PC WORLD ROMANIA, an 5, nr.5, 1997, pag.86-87.
[5] Ion IVAN, Mihai POPESCU, Panagiotis SINIOROS, Felix SIMION – *Software Metrics*, INFOREC Publishing House, Bucharest 1999.
[6] Ion IVAN, Doru CAZAN – *Metricile de calitate ale sistemelor informatice*, Informatica Economica, vol. 8, nr. 3, 2004, pg. 123 – 128.
[7] Ion IVAN, Adrian VISOIU - *Rafinarea metricilor software*, Economistul, nr. 1947 (2973), 29 august 2005, Suplimentul Economie teoretica si aplicata, nr. 466, ISSN: 1221-8669.
[8] Ion IVAN, Marius POPA, Catalin BOJA, Cristian TOMA - *Metrici pentru similaritatea entitatilor text*, Studii si Cercetari de Calcul Economic si Cibernetica Economica, vol. 39, nr. 4, 2005, pag. 43 - 57, ISSN 0585-7511.
[9] Ion IVAN, Catalin BOJA **-** *Analiza metricilor software*, Studii si Cercetari de Calcul Economic si Cibernetica Economica, vol. 40, nr. 1, 2006, pg.65 - 78, ISSN 0585-7511.
[10] Ion IVAN, Cătălin BOJA – Metode statistice in analiza software, Editura ASE, 2004.
[11] Loredana MOCEAN, Robert BUCHMANN, Ciprian-Nicuşor DRAGOTĂ – *The Proceedings of the International Workshop on COLLABORATIVE SUPPORT SYSTEMS IN BUSINESS AND EDUCATION,* October 28-29, 2005, Risoprint, Cluj-Napoca, Romania.
[12] E. HARTONO, C. HOLSAPPLE – *Theoretical foundations for collaborative commerce research and practice,* Information Systems and e-Business Management, Springer-Verlag, 2004.
[13] Mihaela MUNTEAN – *Some Considerations About Portal-Based Collaborative Environments,* The 5th European Conference on Knowledge management, CNAM, Paris, 2004.
[14] Mihaela MUNTEAN – *Knowledge Management in Collaborative Environments,* The 2nd International Conference on Economics an Management of Networks, Corvinus University of Budapest, 2005.
[15] G.C. SILAGHI – *AgentSearch: An Architecture of a Collaborative Multi-agent Web Search System,* Proceedings of the International Conference „The Impact of European Integration on the National Economy", Business Information Systems section, Cluj-Napoca, Romania, 2005.

[16] G.C. SILAGHI – *Mechanisms for Collaboration inside Heterogeneous Multi-agent Systems,* Studia Oeconomica 50, Babes-Bolyai University, Cluj-Napoca, 2005.
[17] G.C. SILAGHI – *About a Naive Algorithm for Producing Collaborative Recommendations,* Proceedings of the International Workshop on Collaborative Systems on Business and Education, Cluj-Napoca, Romania, 2005.
[18] Raluca ARBA – *Collaborative Electronic Marketplace*, in International Workshop „Collaborative Support Systems in Business and Education", Cluj-Napoca, october 2005, pag. 11.
[19] Ovidiu DOBRICAN – *An Example of Collaborative System*, in International Workshop „Collaborative Support Systems in Business and Education", Cluj-Napoca, october 2005, pag. 48.
[20] Vasile Paul BRESFELEAN, Mihaela Antonia BURCA – *Collaborative E-learning Systems,* in International Workshop „Collaborative Support Systems in Business and Education", Cluj-Napoca, october 2005, pag. 17.
[21] B. DAVID, O. DELOTTE, R. CHALON, F. TARPIN-BERNARD, K. SAIKALI – *Patterns in Collaborative System Design, Development and Use,* Laboratoire ICTT, Ecole Centrale de Lyon, France, 2003.
http://dex.francu.com
[22] Duncan STEVENSON, Matthew HUTCHINS, Chris GUNN, Matt ADCOCK, Alexander KRUMPHOLZ – *Multiple approaches to evaluating multi-modal collaborative systems,* CSIRO ICT Centre, Australia, 2005.