THE 3C COOPERATION MODEL APPLIED TO THE CLASSICAL REQUIREMENT ANALYSIS

Vagner Luiz Gava
IPT - Institute for Technological Research of the State of Sao Paulo, Brazil
Mauro de Mesquita Spinola
Antonio Carlos Tonini
José Cardenas Medina
University of Sao Paulo, Brazil

_____

ABSTRACT

Aspects related to the users' cooperative work are not considered in the traditional approach of software engineering, since the user is viewed independently of his/her workplace environment or group, with the individual model generalized to the study of collective behavior of all users. This work proposes a process for software requirements to address issues involving cooperative work in information systems that provide distributed coordination in the users' actions and the communication among them occurs indirectly through the data entered while using the software. To achieve this goal, this research uses ergonomics, the 3C cooperation model, awareness and software engineering concepts. Action-research is used as a research methodology applied in three cycles during the development of a corporate workflow system in a technological research company. This article discusses the third cycle, which corresponds to the process that deals with the refinement of the cooperative work requirements with the software in actual use in the workplace, where the inclusion of a computer system changes the users' workplace, from the face to face interaction to the interaction mediated by the software. The results showed that the highest degree of users' awareness about their activities and other system users contribute to a decrease in their errors and in the inappropriate use of the system.

Keywords: Software Requirements, Computer Supported Cooperative Work, 3C Cooperation Model and Awareness.

_____

1. INTRODUCTION

Due to its nature, cooperative work depends on people`s will to work together; it cannot be prescribed (Daniellou & Six, 2003), and once information systems are introduced, the environment and the activity are thoroughly transformed. In such a case, the work with the new system will differ from the current one and even if the existing work goes through an accurate analysis, there is no guarantee of a possibility to forecast completely its future utilization.

This matter is especially important when there is the production of information systems which must be used in an environment that will replace processes involving cooperative work.

Sommerville (2010) mentions the importance of using alternative methods while finding requisites for an information system with specific regards to the following situations:

- Software requirements come from the way people work (real work) rather than whatever the processes definition recommends for the work (prescribed work);

- Software requirements come from the cooperation and perception of other people`s activities.

In the traditional approach for software development, the most frequently used hypothesis regards models that are centred in one single user (seen as standard and independent on the environment or group in which it is inserted); they are generalized for the study of collective behaviour involving all the users.

In order to be possible to address the paradox of design (Daniellou, 2007) - when developing a computer system that will replace part of an IS, it is necessary to have this system available in advance, and in order to conceive it appropriately, it is essential to be aware about the future activities in the first place;   a process that deals with this matter must be established, essentially considering the fact that people work together to achieve several goals established and also that users have inherent difficulties to discuss effectively on how this cooperation happens (Sommerville, 2010).

The process proposed for this study uses ergonomic techniques, software prototyping, the 3C cooperation model and awareness and classical software engineer concepts with the purpose of dealing with collective and cooperative working issues which must be considered in the project of a computer system. To do so, the action research methodology is applied while the system is found in development.

This article allows us to answer the following question:

- How can the 3C Cooperation Model and Awareness be applied to the elicitation of requirements for cooperative work?

We accept as a premise that conventional methods used for software development do not deal properly with the collective dimension of the work in the information system, neither in its conception nor in its improvements/corrections.

The process is applied in the phase of cooperative workflow implementation in a big company of technological research in Brazil. It shows how to consider the change of the face to face interaction for an interaction intermediated by the software with its respective results.

This article is organized as follows: first of all, the definition of the main concepts to provide the basis for the proposed process, and then these concepts are logically linked according to the proposed process. The results are presented through action research based on the given theory. Finally, they are discussed and we call attention to some recommendations to carry on the work.

2. MAIN CONCEPTS

This section is to define the main concepts underlying the proposed process. It is initially through the conceptualization of collective work, in 2.1 and then, in 2.2 we focus on questions of how to use prototyping. In 2.3, we borrow the the 3C Cooperation Model concept and 2.4 presents the awareness concept and its relation with the 3C model. Finally, 2.5 and 2.6 define model concepts and software requirements, respectively, which were used in this paper.

2.1 Collective work

Once the collective dimension of the work makes the situation more flexible, more complicated and less deterministic (for a given observer) regarding interrelations (interactions, retroactions, interferences, etc.), the systemic complexity increases by manifesting the fact that the whole presents qualities and properties which are not found in the perspective of parts, in isolated consideration. Also because parts have qualities and properties that disappear under the aspect of organizational coercions of IS (Morin, 2002).

A new order of complexity comes when the existence and maintenance of its diversity are not separable from interrelations with the environment, interrelations in which the IS gets outside matter/energy in a superior degree of complexity (which are autonomous and dependent at the same time): information (Morin, 2002).

The cooperation term, which is about dealing with working matters, may be defined in a wider way regarding two or more people cooperating when working on a joint task with a certain objective. In order to accomplish this goal, actions from both parts are necessary in a way that cooperation means common operation with the purpose of having an adjustment through new matching operations as well as reciprocity or complementarities (Piaget, 1996).

The definition used in this paper is given by Dejours (2005, p. 93):

"cooperation is a coordinated conduct defined as the action to join a common task. Cooperation assumes to be somewhere where single contributions converge and dependency relationships among parts get solid at the same time"[1].

The author highlights that cooperation refers to collective work and it is a coordinated conduct that enables superior and supplementary developments in relation to individual developments.

Cooperation does not idealize the human operator because it regards the integration of differences among people articulating each worker`s specific abilities and compensating possible single flaws. According to the author, reliability, safety and the job quality are directly linked to the quality of cooperation that may be able to

---

[1] TN: My own translation.

compensate failures concerning the organization of a prescribed job and restrictions of human developments.

The individual integrated to an IS, in which there is a distribution of competences, tasks, and roles, needs integrating processes (coordination, communication, organization/cooperation). The duality between the whole and the parts, between unification and distribution, between homogeneity and heterogeneity are rarely taken into account for methods of analysis and conception of computer systems (Erceau, Chaudron, Ferber, & Bouron, 1994). This case is much more detailed in item 4, where integrating processes will be associated to cooperative work of a given group and focused by implementing software.

2.2 Prototyping

When developing software, a prototype corresponds to a version of the system which is available right in the first steps of a development process. Functional prototyping, according to Boar (1985), implements parts of the system requirements through the construction of a prototype that performs a real behaviour of this system (by algorithm implementation and databank) which can also use tools especially built for the production of this kind of prototype (functional prototyping is used in cycle 3 of this research).

Afterwards, this prototype is discarded for the next step which is the effective development of the system in accordance with a traditional sequence (analysis, project, implementation and tests) holding a set of well refined requirements.

This work employs the term of incremental or evolutionary prototyping as a synonym of incremental development (the prototype is not discarded but it evolves to reach stakeholders` requirements).

On the other hand, the non-functional prototyping (used for cycle 2 of this research) attains the stakeholders` behaviour and the system by interactions and interactions by them through a set of graphic interfaces simulating the real behaviour of the system (without the algorithms and databank implementation).

The use of system prototypes (functional or not) provides several advantages (Boar, 1985; Kotonya & Sommerville, 1998; Leffingwell, 2003):

−Distance reduction among the project participants: communication is a crucial development problem . Even when someone knows what they want, there are always changes whenever these needs become requirements;

−Increase in the agents` participation and interest: complex systems which involve several areas of one company claim for commitment, agreement and consensus among several agents in order to have them operating correctly;

−Permission of measures according to the size of functionalities through function point analysis: from the function identification of a given type and transaction type and also a sketch of the system data model, it is possible to calculate function points, from the very beginning of the system and its refinement during several development cycles;

−It is a vehicle to validate requirements which enables to test its interfaces from early stages.

Regarding the user interface, according to Pressman (2009), prototyping is the only practical way to validate whatever has been projected.

2.3 The 3C Model

2.3.1 Introduction

CSCW (Computer Supported Cooperative Work) is the field of study that investigates how people work together using computer technology. Typically, CSCW applications include e-mail, videoconference, chat systems, interactions among multiple individuals, shared applications in real time, notification systems, and support perception.

Groupware is a supporting technology for the interaction among participants of a working group and normally considered a synonym of CSCW. This technology has been a lot diffused to model distributed systems using digital media and computer nets (Fuks, Raposo, Gerosa, Pimentel, Filipo & Lucena, 2011).

The 3C cooperation model used in this research comes from article Ellis, Gibbs and Rein (1991) and is supported by the conception that to cooperate, members of a group (C) communicate, (C), coordinate and (C) collaborate (3Cs). As we can see Figure 1, there is a cycle indicating that people must communicate to coordinate their working efforts and collaborate for one single objective.

For cooperation, communication is necessary; it can be directly or through information attained within the environment where the work takes place. There is stimulus provided by awareness information (2.4) in each relationship and they enable a sharing understanding to happen around the collaboration object. The object means goals and objectives established for a task conclusion or even for the whole job (Assis, 2000).
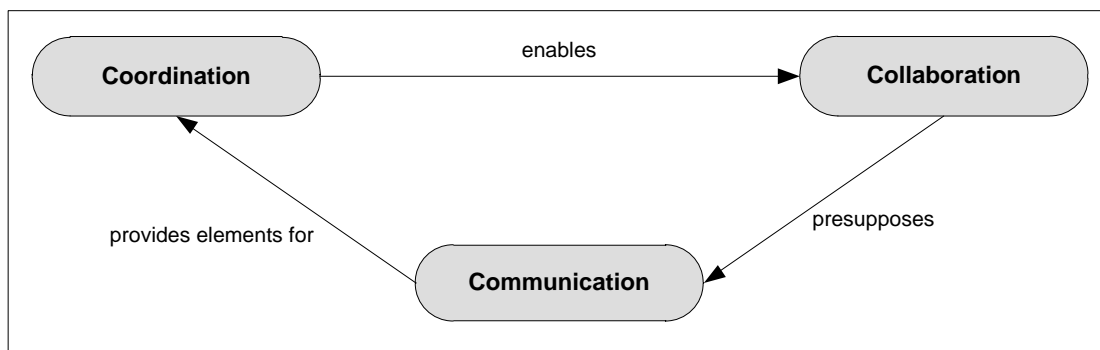
Figure 1 - Cooperation models
Source: Ellis *et. al* (1991) & Fuks, Raposo, Gerosa, Pimentel, Filipo and Lucena (2007)

Despite the separation of these activities for analysis purposes, communication, coordination and collaboration are not carried out in tight and isolated way; they are performed continuously and iteratively while the group work takes place (Fuks *et al.,* 2007). Tasks come from commitments dealt with during the communication process, managed by coordination and carried out during collaboration. With the awareness mechanism (2.4), the individual gets feedback from their actions and feed through their co-workers actions.

Cooperation is a joint operation of group members in a shared space that performs tasks when generating and manipulating collaboration objects while tasks are

being carried out. At the collaboration time, it is necessary to renegotiate and make decisions concerning unexpected situations and this requires new rounds of communication and coordination.

Before the task being effectively performed, the group gets organized and articulated, for instance. For such an activity there are also specific needs of cooperation; they are different from necessities that happen during the task execution. Individuals who plan it may not be the same who perform it as it normally happens in assembly lines where activities are planned and later individuals performs their tasks without any kind of interaction with one another. In cooperation, the plan is dynamically renegotiated and it is not possible to separate thoroughly coordination from collaboration. In the course of cooperation among individuals, they learn and refine working processes renegotiating initial plans and intercalating actions and negotiations (Gerosa, 2006).

2.3.2 Instantiation of 3Cs models in cooperative systems - CSCW

Here we will present 3Cs interrelations that illustrate how different application domains lead to different combinations of cooperation dimensions in which their respective cooperative systems need somehow to reflect in their projects. In this section, three types of 3Cs instantiations (occurrences) will be presented and the instantiation for other kinds of CSCW systems is possible, as for example, networking websites, media spaces and family calendars (Fuks *et al*, 2007)

The first instantiation of the 3C model focuses on the group work domain (Figure 2) in which conservation is targeted to action: as communication happens, people negotiate and make decisions; while coordination happens, they deal with conflicts and organize their activities in a way that they avoid a waste of communication and efforts of collaboration. This necessity of decision-making and renegotiation regarding unexpected situations that happen during the collaboration process demands a new round of communication which will therefore demand coordination to reorganize tasks to be performed all along the collaboration process. This is the case of a service debate.
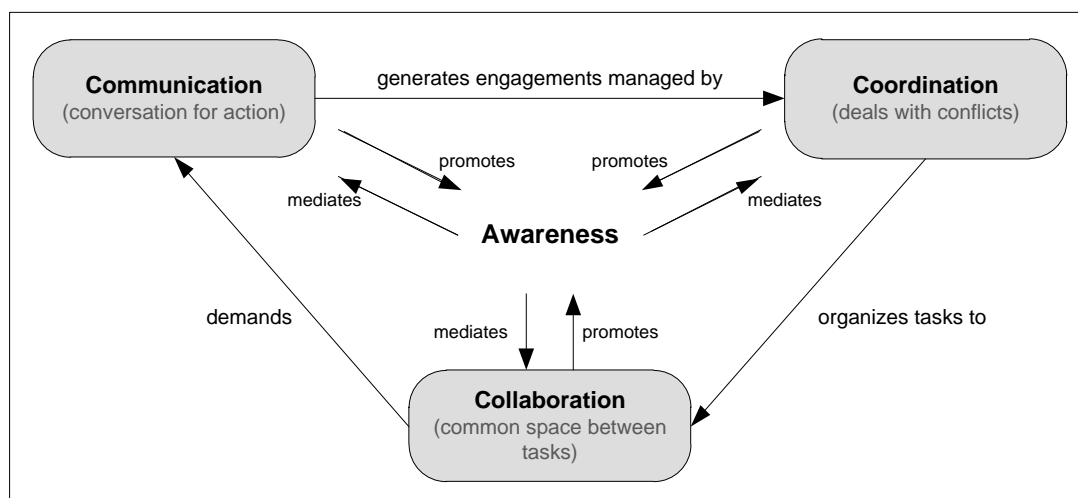
Figure 2 – 3Cs interrelation in conversation for action
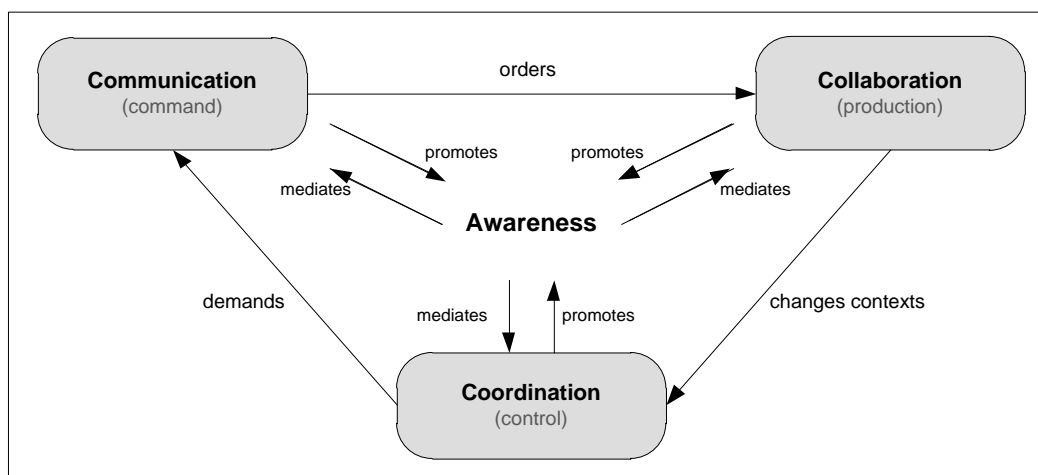**Source: Fuks *et al.* (2007)**

Figure 3 shows a workflow. When performing pre-defined tasks by a workflow, the group members feel they need to change whatever was previously arranged. A counterexample is the classic industrial assembly line where employees are not expected to negotiate execution in respect to next tasks based on attained results of previous executions.
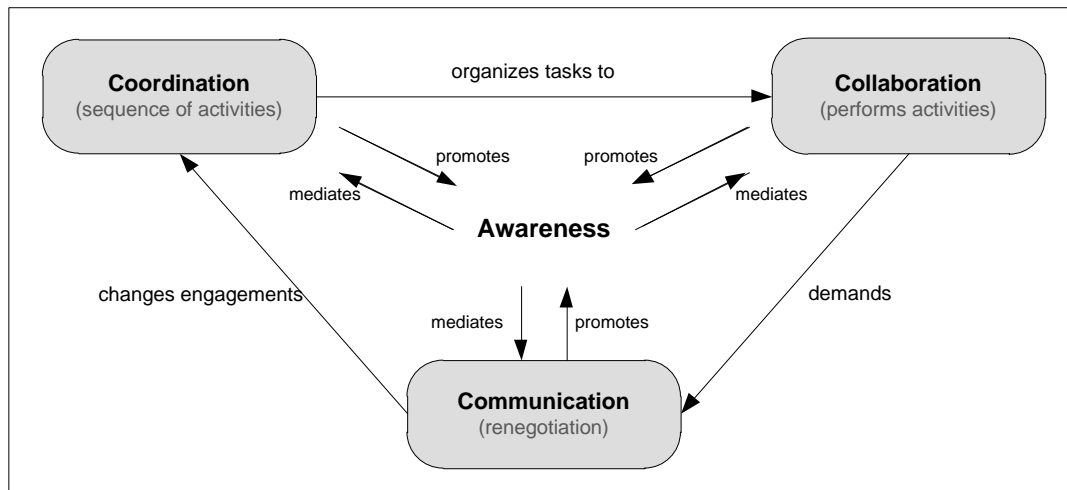
Figure 3 – 3Cs interrelations in adaptive workflow systems
Source: Fuks *et al.* (2007)

Finally, another instantiation about the three dimension of the 3C model is presented; it characterizes command and control (Figure 4). Here coordination is a supervision fact that inspects the group production and command changes. An example where command and control are collapsed is in the oar where the captain observes the development of a rower group and requests for more effort.

Figure 4 – 3Cs interrelations in Command and Control
Source: Fuks *et al.* (2007)

These cycles show the interactive nature of cooperation. They also highlight the 3Cs interactions among themselves. These interactions are mediated through awareness information available in the environment itself, and they are consequently influenced by each dimension of cooperation (Fuks *et al*, 2007).

2.4 Awareness in cooperative work supported by computers

2.4.1 Introduction

While the interaction between people and the environment within a face-to-face situation seems natural and considering senses as vision and hearing fully available, the situation gets less clear when there is a trial to provide support to perception in the virtual environment (Fuks & Assis, 2001). These environments tend to hide several pieces of information that would be available in a face-to-face meeting.

Another aspect to be acknowledged is the relation among people within a cooperation environment. This environment must facilitate interaction understanding among participants in a way that individuals manage to comprehend events and foresee possible necessities. The environment must also enable interactions to happen in an appropriate way; this means that information must be provided in an organized way to prevent erroneous data interpretation (Assis, 2000).

Relevant perception information must be seen, how it can be generated, where perception elements will be necessary and in which way these elements should be presented. It must be carefully considered that the elements really assist cooperation instead of making it more difficult.

2.4.2 Awareness

For this paper, awareness is defined as consciousness about the contextualization of individual activities through comprehension of activities performed by other people (even when they are not directly communicating); it refers to being acquainted with group activities, to knowing what happened, what has been happening and/or what might happen besides the knowledge about the meaning of this work and the group. In summary: awareness means comprehension about the total status of the system including past activities, current situation and future options (Brinck & Mcdaniel, 1997; Pinheiro, Lima & Borgwe, 2001; Santos, Tedesco & Salgado, 2011).

This awareness is crucial for coordination with other individuals in cooperative tasks where there is not always a direct communication and it can even refer to indirect ways of communication as, for instance, deductions or suppositions about what the other person is arguing based on transmitted information or used gestures in the space they share.

It is a complicated situation for distributed computer systems (used by several users in different kinds of environment). Resources for this kind of information are poor if compared to resources of a face-to-face scenario whose interaction mechanisms are different than the usual ones. As a result, working together with software intermediation, which is based on digital and distributed technology, apparently may seem inefficient and weary if compared to face-to-face work. This is an important factor of awareness for the cooperation flow and normality (Santos, Tedesco & Salgado, 2011).

2.4.3 Awareness and 3C model

This section aims to draw a parallel between the necessity of attention and science in order to promote a task understanding of a piece of information or an objective inside the working environment as well as fundamental concepts of the 3C model groupware.

Awareness and communication

Communication may be understood as a mean to share information. It can happen in distinct levels and different ways. The relevance of information established between the communicator and receptor depends on the context into which it has been inserted and on inherent rules for the establishment of communication, appropriate means, pertinent information, and other factors (Calvão, Pimentel & Fuks, 2011).

Synchronous communication is understood as communication in real time where stimuli are noticed almost at the same time they are accomplished; asynchronous communication is where information is not noticed at the same time it is transmitted (Santos, Tedesco & Salgado, 2011).

When cooperative work happens through software, there is a change in the face-to-face scenario for communication, anywhere and anytime. Besides, there is the creation of a new and more generic domain compared to the conversation one; the communication domain presenting new vehicles for information transmission and new contexts for knowledge sharing.

Structures of a language used in this communication must be present for the understanding of the transmitted content. To do so, however, the knowledge of using new media is necessary and they become part of the communication domain.

Awareness and coordination

It is crucial that each member is aware about the working progress of their colleagues when decisions and results depend on the integration of efforts from different group members, as for example: whatever is necessary for an accomplishment, preliminary results, etc. Thus, an information system that supports cooperative work must provide information to its participants about what must be done and actions the colleagues have been doing.

The term awareness information is used to describe this monitoring of activity progression or group production constituting one context for individual work whose comparison are activities of the rest of group (Dourish & Belloti, 1992). This context is used to guarantee that contributions established are appropriated for group activities and the process of cooperative work. Without this context, individuals cannot measure the quality of their own work in respect to the group objectives and progresses.

Awareness information must be also introduced whenever someone needs to know what must be done in order to proceed with their work; this kind of information is not always associated to working knowledge about the work of the rest of the group. In this case, it is necessary to indicate which steps should be taken through some communication mechanisms or through the availability of some repository that holds this kind of information in the environment.

Thus, awareness channel implicit to this mechanism must offer some written or spoken information, some icons, diagrams, photos, etc. Awareness information generated by communication within a working environment will be useful for group coordination if it is presented in a clear and ordered way.

Besides of comprehending exchanged information, members of a working group need to be engaged with the work. This means that actions must be coordinated in a way that there are no losses in the communication effort; this would make cooperation more difficult. Shared knowledge of a certain situation will happen if the working group

presents equivalent expectations related to the actions that must be carried out (Fuks & Assis, 2001).

Expectations are interpretations of the necessary awareness information for an understanding of the situation and for some evaluation like the development of the established goals of the cooperation object.

There will be new ways to manage information in virtual working environments. Man-computer interaction must be taken into account when the data presentation to be evaluated by users is happening as well as during all the conception of this environment where there are several elements that must be regarded for this interface construction as much as the construction of utilities towards this interaction (Assis, 2000).

Awareness in collaboration

Interaction among individuals or between one individual and objects of some working environment does not consist of a collection of activities which are not related. There is a set of new events as a result of the interaction that involves a set of information which consequently generates a cognitive structure that individuals seek for knowledge in a way of planning and coordinating future interactions.

Awareness develops functions in different levels under the collaboration point of view. In a more abstract level of a hierarchical scale of awareness, perception over the characteristics of other participants` activities enables individuals to structure their activities, then avoiding work duplication. In a more factual level, there is the perception over the activity content of each person turning the work into a more accurately distributed and shared one. Also, there is some synergy in the group behaviour in respect of the collaboration object; these factors must be present in cooperative applications.

The space of shared interaction is necessary for shared understanding. This space is provided by tools and artefacts for the information movement and they facilitate the contact among users (Santos, Tedesco & Salgado, 2011).

2.4.4 Awareness elements

Awareness elements are elements of shared space where information to provide awareness is available (perception). Information systems developed for cooperative environments must provide awareness elements that make the necessary information to cooperation available in an adequate way as much as in an individual work.

Getting to know how communication, coordination and cooperation mechanisms work, and mainly how they must be used to keep different awareness elements, the designer of the shared virtual environment may create techniques and tools that provide appropriate information for users regarding goals, tasks and other elements of the environment.

Guided by their own perceptions, individuals create a shared understanding and get coordinated in a way that their individual efforts add value to the groups` work. When projecting an information system with such characteristics, there must be taken into consideration which pieces of awareness information will be necessary considering how to generate, gather and distribute them (Gerosa, Fuks & Lucena, 2003).

Awareness information is relevant for both group and individual work. Some examples of awareness information, which are more suitable for individual work, are about the pieces of information from messages the individual has already read and what

The 3C Cooperation Model applied to the classical requirement analysis

is new since his/her last visit. On the other hand, group-oriented information is present in the environment of people working with an artefact, for example. Despite this separation, all kinds of information must be projected in order to complement each other and to assist individual work in the cooperation context. Other examples of useful awareness information to contextualize participants are the following: common objective, each person`s role within the context, what must be done, how to proceed, what impacts the actions cause, where to act, who is around, what the partner can do, what the other people are doing, location, origin, importance, relations, authorship of cooperation objects (Gerosa, Fuks & Lucena, 2003).

For awareness support, some considerations must be taken into account: what information to provide, how to predict it and how to give individuals the information control (if it can be viewed, changed, etc.) (Brinck & McDaniel, 1997). Types of awareness information in groupware vary as followas: perception about documents, projects, tasks, and even about the colleagues` location and work activities that may be presented in a written or spoken way through photographs, icons, etc.

There are several types of awareness elements; they are classified by purpose, time, scope, abstraction, aggregation, perspective, provision way, personalization, and others (Brinck & McDaniel, 1997). These elements basically aim to respond the following questions: "who, what, where, when and how". These questions must be asked all through the environment seeking for which elements users should be conscious about in relation to the situation and to provide understanding.

Table 1 and Table 2 show elements characterized by their meanings and they are divided into two tables: Table 1 to characterize present events and Table 2 to characterize past events. The elements are all part of common knowledge and deal with interactions between people and the environment.

Table 1 - Awareness elements related to the present

Category	Element	Meaning
Who	Presence	If there is anyone present in the environment
	Identity	Who is taking part of the activity
	Authorship	Who has performed a particular event
What	Actions	What must be done and what the others are doing
	Intentions	What the objective of an action or work is
	Artefacts	In which objects individuals are working at the moment
	Production	What the preliminary results of some work are
Where	Location	Where people are found in the environment
	Observation	Where people are looking at
	Vision	To where they can or should look at
	Range	Up to where people can or should perform

Source: ASSIS (2000)

Table 2 - Awareness elements related to the past

Category	Element	Meaning
How	Action background	How operations happen
	Artefact background	How a particular artefact reached that state
When	Event background	When a particular event happened
Who (past)	Presence background	Who was in one location of the environment and when
Where	Location background	Where a particular individual was
What	Action background	What a particular individual was doing
	Production background	Which experiences are materialized at work

Source: ASSIS (2000)

Having this data, we can evaluate, for instance, how face-to-face situations would be translated into a groupware environment. This does mean that support should be equally given to all elements in the interface.

There are two crucial factors to determine how each element should be dealt with. The first one is which interaction degree among participants of one activity indicates how general or specific information must be provided. The second deals with the element dynamism: "how often do elements change?" This factor indicates, in a general view, the number of times that the interface must be changed to reflect new pieces of information.

Some particular elements will never change in some situations; that is the reason that there will not be the necessity of explicit support in the interface. If some activity always happens on the same day and time, there is no need for the system to accumulate and to distribute detailed information about the actions that must be taken.

Other information type to one specific objective will happen. In spite of that, this basic set of elements provides a high level organization to facilitate perception in shared environments. Elements related to the present are characterized by information that must be dealt with and normally distributed as feedback. Besides this characteristic, elements related to the past must be kept for future treatments (Assis, 2000).

2.5 Software model and process

A software process is an organized set of activities and results associated that transform inputs and outputs and generate a software product. A software model process is a simplified description of a software process, a useful abstraction to explain different ways to approach the development (Kotonya & Sommerville, 1998; Pressman, 2009).

In the evolutionary approach of iterative development model, there is a system developed through successive versions. An executable system is quickly generated based on initial specifications. Then, it must be refined in accordance with the client`s attained feedback with the aim of producing one system that will be satisfactory to his/her needs. So, the system is delivered - one alternative is to re-implement it - with the use of a better structured approach in order to produce a stronger system with more maintainability.

There are two main evolutionary development strategies:

–   Discardable prototypes.   The purpose of building discardable prototypes is to define requirements which are misunderstood with the aim of developing a good specification. In this case, prototyping is focused on a requirement definition which has been poorly defined;

–   Exploratory development (evolutionary model). The development starts with well-defined parts of the system and it will progress according to the increase of new characteristics required by the client.

According to Sommerville (2010), for small and medium systems, the best solution is an incremental solution. On the other hand, for complex, large and long-term systems or even for systems developed by different teams, the best solution contemplates prototyping use (discardable or not) to define misunderstood requirements with an implementation through a better structured model (cascade model).

This work, according to Sommerville (2010), employs the term of incremental or evolutionary prototyping as a synonym of incremental development whose prototype is not discarded but it evolves to reach stakeholders` requirements.

2.6 Software requirements

For Sommerville (2010), requirements are description of how the software should behave, information on the application domain, restrictions about software operations or property specifications or software attributes. Requirements are defined during the first steps of the software development with specifications of what might be implemented.   Requirements invariably present a mixture of problem information, explanations of behaviour and software properties, project conditions and construction constrains.

Software requirements are classified as:

–   Functional: they must describe what the system must do, how it should react to particular inputs and how it should behave in some given situations; therefore, they refer to input and output conditions and demands of transformation.

–   Non-functional: they are service or function restrictions offered by the system. They include time restrictions, and process and standard restrictions (Souza & Castro, 2004). They are generally applied to the system as a whole and may be classified as: Product and external process requirements. They regard technical specification of standards and methods about the productive process, product quality and desirable characteristics, and applicable policies for the process and product generated;

–   Domain requirements: these requirements come from the application domain and they reflect on characteristics and restrictions of this domain; they may be functional or non-functional. They usually include specialized terminology in the application domain or reference to its concepts. These kinds of requirements are important because they reflect on the grounds of the application domain in a way that it may be impossible to develop the information system if these requirements are not satisfied;

–   User requirements: they must describe functional and non-functional requirements in a way they are understandable by the users with no need of a detailed technical knowledge. They must only specify the external behaviour of the system

avoiding the most technical details about it and focusing on key facilities that will be provided;

– System requirements: it is a wider version of the user requirements which are used by system developers as a starting point for the system project. These requirements add details to explain how user requirements may be served by the system and must focus on the external behaviour of the system as well as its restrictions with no concerns on how the system will be implemented or projected.

The requirement sources (stakeholders, domain and system) may be represented by the system viewpoints because each viewpoint represents a subset of the system requirements (Sabetzadeh, Finkelstein & Goedicke,2010);

3. RESEARCH METHODOLOGY

This research uses the action-research methodology (AR) that corresponds to one kind of social research with empirical basis conceived and carried out in aclose association with an action or with the resolution of a collective problem in which researchers are involved in a participative or cooperative way (Thiollent, 2011).

This action-research reaches three main phases: preliminary, driving cycle and metaphase (Coughlan & Coghlan, 2002).

The first phase (preliminary phase - Figure 5) is about a context understanding in which the research will be carried out as well as the purpose of conducting the work. This phase also involves a justification establishment for the required action besides justifications for the research.
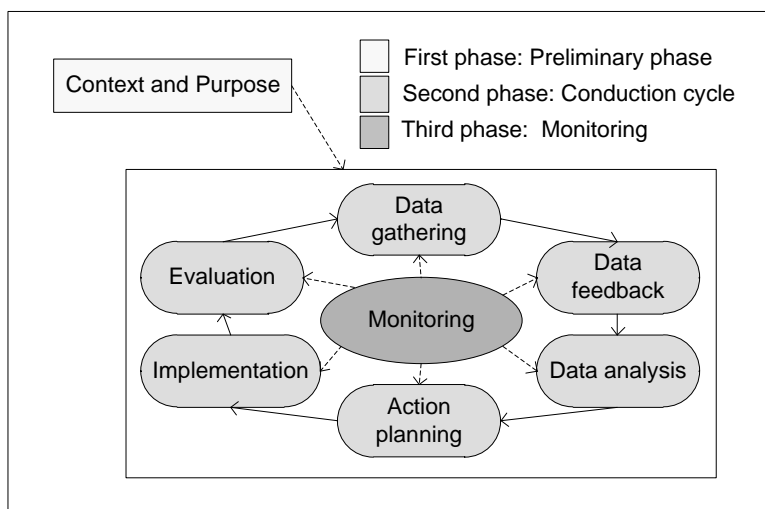
Figure 5 – Action-research cycles
Source: based on Coughlan & Coghlan (2002)

The second phase (Figure 5) is composed of six steps and it starts with data gathering (diagnosis and/or data collected when the research is found in progress), data feedback (for those who are involved in the research), data analysis (with those who are involved in the research), action planning (definition of future intervention), action implementation (to put everything that was planned into practice), and evaluation (to

check whether the implementation results will bear desired results or not) returning to a new data gathering (if necessary) and closing the cycle (first interaction).

The third step (monitoring - Figure 5) reaches the inspection of each of the previous six steps in a way of controlling whatever learning is generated during action-research with its monitoring being performed in different ways as the research advances through the steps of the second phase.

Research lineation

With the purpose of complying with questions and objectives posed in I.2, this research uses the action-research methodology (see item 3) through driving cycles (Figure 6) and the proposed process (see item 4) with the following planning:

1.Methodological and applied literature review (that refers to cooperative working concepts, Requirement Engineering and related issues);

2.Context and purposes: production of a specific context of AR;

3.Conduction of the first action-research cycle: process to specify software requirements focused on the identification of individual characteristics at cooperative work and domain characteristics;

3.Conduction of the second action-research cycle: process to specify software requirements focused on the identification and simulation of cooperative characteristics at work;

3.Conduction of the third action-research cycle: process to specify software requirements focused on refining characteristics of cooperative work (in real use);

6. Production of an article with research results.
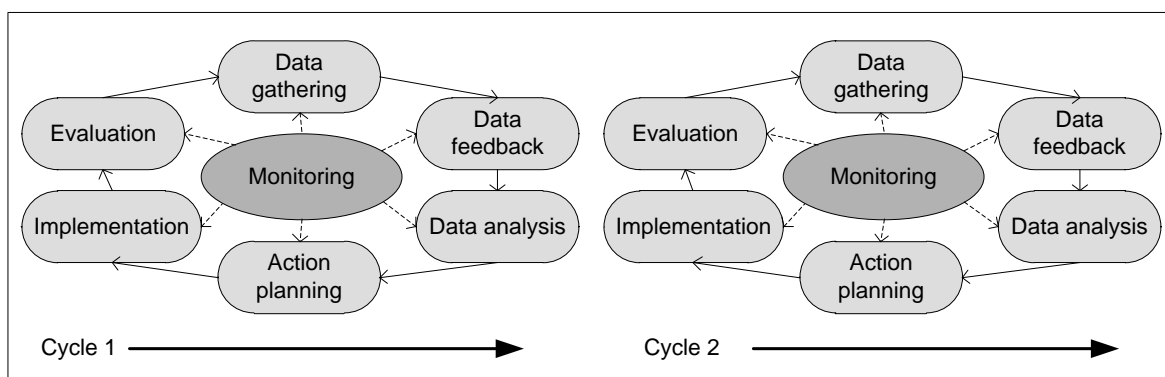
Figure 6 - Interaction of Action-research cycles

The main focus of this article regards the conduction of the third cycle of this action-research from the 3C cooperation model perspective. Cycles 1 and 2 are better approached in Gava, Spinola, Gonçalves, Medina and Tonini (2011b) and Gava, Spinola, Medina and Tonini (2011c).

Action-research cycles

The final objective of the proposed process is to attain the system requirements (item 2.6) and models (which is the document the software developers must perform);

this means that the expanded version of users` requirements used as a starting point for the system project is the document that software developers must perform.

In systems which one evolutionary solution is adopted, this document can be simpler by focusing on users` requirements definition and high-level functional requirements (Sommerville, 2010).

Action-research context

This work presents results of an action-research carried out while some cooperative workflow software was being developed in a Brazilian technology company.

One of this company`s mission is to provide technical support to productive sectors and this is done through laboratory services like the development of technical advice for areas of civil engineering, metallurgy, mechanics, industrial electricity, data processing, higher education and training.

At this moment the company owns 40 laboratories that offer services, and, despite of a set of internal norms about general aspects that must be performed for the service of a particular request (budget, etc.), each of these laboratories apply these norms in their particular way because the service is not centralized.

As a result of this lack of standardization about the information about service procedures that cause independent ways of performing in each laboratory (several of them presented service procedures through paper file), information are fragmented and the company presents difficult aggregation even to come back to the service situation of a particular client.

On the other hand, this service procedure corresponds to workflow which passes through several steps, from request opening to its accomplishment in a way that involves the laboratory participants, who work collectively (technicians and secretaries), especially at laboratories that offer several kinds of service that complement one another (the same request may need several of these services from the same laboratory).

Thus, having this reality presented, this company`s directorship approved the development of an information system with the following objective, taken from its vision document:

"The laboratory monitoring system aims to standardize monitoring and managing laboratory service methods through all the company in order to offer homogeneity and more efficiency to current technical service development and monitoring, from the first moment of a service request to its last, the billing. The systems can generate budget, register samples, have number of technical documents, and create billing requests in a thoroughly integrated way. By a set of reports it is also possible to monitor daily activities in the laboratory as well as to provide management information about the main performed activities."[2]

Thus, this computerized system follows the several steps of service requests, from solicitation opening, through all execution phases to its accomplishment; this way it involves collective work from all laboratory participants (technicians, supervisors and secretaries). The standard sequence of each service (responsible and tasks) is previously defined (see Figure 10) but it may change depending on each case, for example, when

---

[2] TN: my translation from Portuguese.

exchanging the staff`s temporary functions or special service situations that are not programmed in the information system.

Action-research conduction: procedures to simulate cooperative work from the computerized system perspective

The procedures to simulate cooperative work were the study the object of the action-research cycle 2. In spite of the fact that this step of the research is not the objective of this article (action-research cycle 2), this cycle is here presented in a summarized way with the aim of facilitating cycle 3 understanding which is, in fact, what this work focuses on.

Thus, this conduction cycle was applied to a process to simulate and identify cooperative characteristics of the job whose main purpose is to attain transactional characteristics of cooperative work by using the Collective Analysis at Work (CAW) techniques described by Ferreira (1993), mental model and interaction (Norman, 2006) and software models and procedures anchored by artefacts generated during the conduction of this cycle.

As it has been presented in 2.2, we opted for non-functional prototyping technique with the development of successive software prototypes offering a common representation to communicate with users and designers and also constituting a guide to specify successive versions. Prototypes were presented to users for collective discussion using the CAW and the starting point was the user`s working baseline from their own point of view to project typical actions of the future computerized system.

The main objective of this cycle is to have requirements of used systems as a starting point for the software project (third cycle) highlighting the definition of users` requirements and high-level functional requirements (once an evolutionary solution is adopted) focusing mainly on the evolution of cooperative work requirements.

The conduction of the third action-research cycle that corresponds to the process to specify software requirements focused on refining characteristics of cooperative work (in real use) is better detailed in 4.3.

4. PROPOSED PROCESS

4. 1 The process general view

We can see a simplified diagram of the proposed process for this research in Figure 7. The highlight refers to a cut-out and we intend to focus on the development process of information systems within software engineering: the sub-area that deals with requirement engineering (item 2.6).

Within this cut-out, processes to analyse feasibility and applicability are considered to identify individual characteristics in cooperative work, to identify and simulate characteristics in cooperative work, and finally to refine the identification of characteristics in cooperative work (which regards, in fact, the functional version of the system).

First of all, in order to identify individual characteristics in cooperative work, it is necessary to analyse the project viability and to verify if the system is a good candidate for the application of the proposed process, which means that a cut-out must be done to evaluate the process applicability.
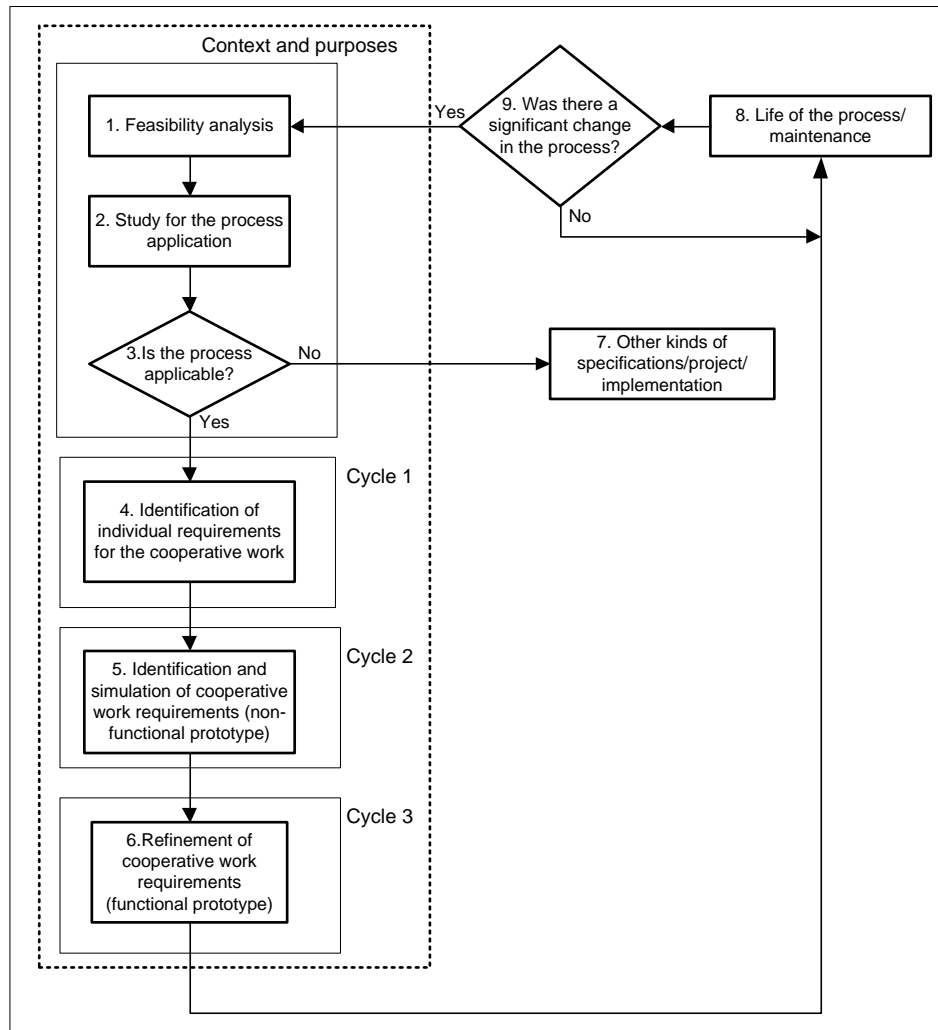
Figure 7 - Main process to identify characteristics of cooperative work. Source: produced by the author

The developed artefacts in the process to identify individual characteristics of cooperative work will be the starting point to study characteristics of cooperative work in information systems. Once necessary requirements are attained, the next step is the system implementation (functional prototype) from where the cooperative requirements of the system will be complemented mainly focusing on the 3C model and awareness concepts.

4.2 Checking the process applicability for the candidate system

The purpose of this step is to check if the proposed theory can be applied to the system or if any other kind of process will be necessary for the development of requirement definition.

As defined in item 2.2 and taking into account item 5.1, prototyping may be applied to a set of candidate systems that must present the following characteristics:

−   The system has a lot of interaction with users through transactions with reports associated to databank and does not operate with a lot of batches (Boar, 1985; Kotonya & Sommerville, 1998);

−   Coordination distributed in users` action (without a defined centre of

coordination - item 2.4),

– Communication among users carried out in an indirect way of data inserted in artefacts of collaboration (there are no specific tools – hardware/software- for direct communication among users - item 2.3),

– Asynchronous environment (users do not need to be working simultaneously in order to reach the objective - item 2.4),

– Uncoupled graphic user interfaces (they are not coupled to interfaces of other group participants - item 2.4),

– Although the phases of the process are normally associated to specific roles in the group, they may be performed by any participants (item 3.3.1).

The next step starts when all sufficient factors are weighed to decide if the proposed process is the most suitable for the given problem.

4.3 Process to specify software requirements focused on refining characteristics of cooperative work - Cycle 3

Activities regarding the last phase of the process amount to the case of identification process and simulation but now there is a real system that has been developed using the main requirements of the previous process.
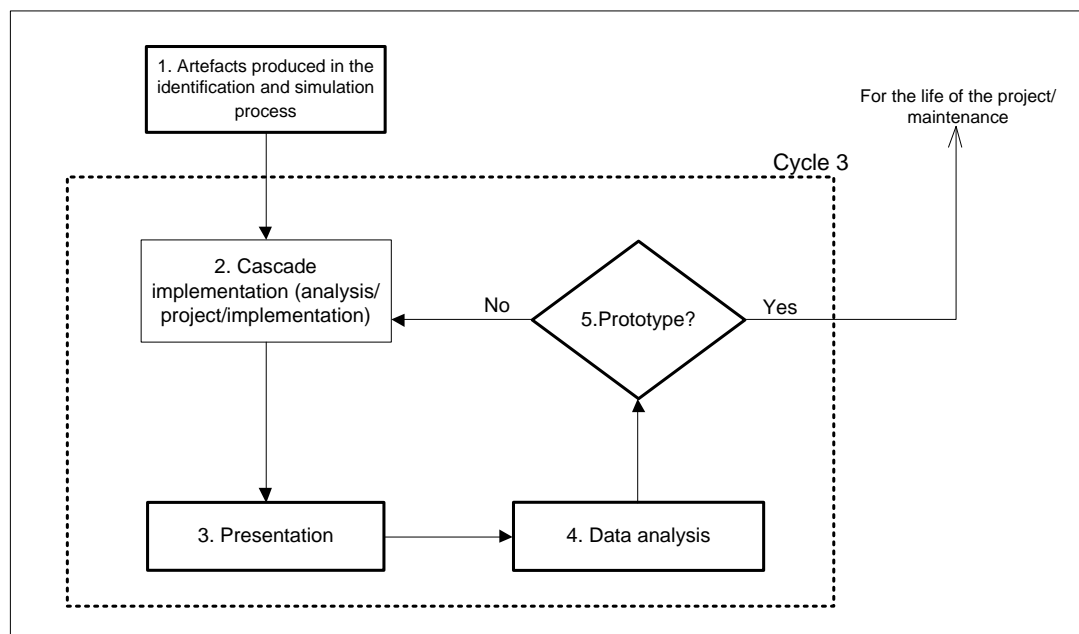
Figure 8 - Process to refine the identification of characteristics in cooperative work.
Source: produced by the author

The phase of the proposed process starts with artefacts developed in the previous phase (second cycle) and used as input in the implementation activity of Figure 8 below. Regarding the previous phase, defined requirements in this current phase will effectively be implemented and not simulated in a preset action/response.

The CAW sessions that will be carried out with users follow the same guidelines defined in the non-functional prototyping process (see item 4.3.2) which uses as an initial physical model the implemented system and gains new components while

interactions and iterations take place in this phase of the proposed process (Figure 8). A distinctive and important aspect for this moment of the process if compared to the previous one is that these sessions will be oriented by the 3C model and awareness concepts.

The presentation of a functional prototype is another distinguishing aspect and must be realized after the systemis sufficiently employed in the users` working environment.

4.3.1 Cascade implementation

Evolutionary prototyping starts its first cycle in this activity; it receives artefacts from the previous phase that will be useful for the development of a first functional version of the system that will be handled by users in their jobs.

In this phase, implementation makes use of cascade model which outputs "flows" from one step to another and the development only carries on whenever one step gets accomplished. To assume such an option in case some changes regarding requirements after phases of presentation and data analysis happen, it is necessary that they wait for the next cycle to be implemented, i.e., there will be an interruption of current requirements discussed with the users during the current iteration.

Gonçalves, Gava, Pessôa & Spinola (2005) present some more details about the architecture used in this activity (it does not take part of the scope for this research).

The system development is oriented to the process flow; for each phase defined as workflow (see figure 10) one or more interfaces were associated and for each of these interfaces the hierarchy of sub-activities and their respective interfaces were associated.

In order to reach the users` mental model we follow the same patterns of the previous process (cycle 2); our starting point is the functional model developed in this initial activity and it will be the initial interaction model of the system.

The main output artefacts are the following: domain requirements, flowchart and interfaces as well as associated functionalities, model of implemented data, navigation diagram and artefacts associated to the 3C model and awareness.

4.3.2 Presentation of evolutionary prototype (functional)

As the case of non-functional prototyping (second cycle), developed artefacts in the phase of cascade implementation (graphic interfaces, interactions, programmed responses, navigation among form hierarchies and flowchart of workflow) will be used as "guides" for the Construction Analysis Workgroup application.

To carry out the first CAW session, the users need to use the system in a real working situation after receiving suitable instructions about the first functional version of the system (first iteration cycle).

In a general view, aspects approached in the CAW sessions (Ferreira, 1993) must focus on the real use of the system regarding the fact that some of these aspects must consider the contextualization of individual activities through an understanding of activities carried out by other people.

–   In order to explain what must be done, it is necessary to ponder about one´s attitude; it is not a common process; it is a way of turning all automatic actions into conscious and explicit ones once persons express themselves and become aware of them; they do not express themselves aiming at other people, they express themselves

in order to be conscious about their own purposes. It is not usual to think about the activity that is done; however, regarding the results, it is the activity itself that is important, it is the one that shall be explained.

− Check what is common and what is different in the activity workers describe with the purpose of evaluating the main detached points and a more detailed characterization of certain aspects of the user`s activity;

− Try to understand the relations of users` activities with other activities: explain what other people do before and after them during the productive process in the hierarchical scale;

− During the session, check the necessity to introduce elements of the 3C model and awareness according to concepts presented in items 2.3 and 2.4 and Tables 1 and 2 in a way of trying to be aware about the contextualization of individual activities through the comprehension of activities performed by other people and in a way the computerized system will reflect on these elements.

4.3.3 Data analysis - evaluating the conclusion of the evolutionary prototype

Data from the prototype presentation in real time are analysed according to the responses given to questions from Tables 1 and 2 with the purpose of achieving awareness elements described in item 2.4.4 and the 3C model described in 2.3. Due to obtained results, suitable elements are selected for implementation and a new CAW session should happen with interrupted requirements over the research-action cycles.

During this phase of the process with systems in use and before the sessions, it is necessary to take interviews with the users who represent the several roles of the workflow aiming to attain necessary information to build and/or correct software artefacts in use.

When this process is accomplished, a report must be written by the researchers and, before its release, it shall be presented to participants so they can detect interpretation errors and other points which were not clarified in the meetings. This confirmation can be also achieved by the users` agreement while CAW sessions take place or after confirming this phase (data analysis).

5 RESULTS

The objective of this research is to have requirements and models of the systems used as a starting point for the software project highlighting the definition of users` requirements and high-level functional requirements (once an evolutionary solution is adopted) focusing mainly on the evolution of cooperative work requirements. Below, the main attained results are presented:

5.1 Dynamic for the 3C model and awareness application

The knowledge of communication, coordination and cooperation mechanisms and mainly how they must be used to keep different awareness elements enable the creation of techniques and tools that provide appropriate information for users regarding goals, tasks and other elements of the environment.

Figure 9 below corresponds to the 3C diagram used in cycles 2 and 3; in order to make coordination and cooperation possible as whole, information about whatever is

happening and what other people are doing is necessary. Through this information, participants can have a shared understanding about cooperation objectives and the goals of tasks or the whole work.

This figure also shows that the starting point that feeds this diagram is the objective of the group, which means the work is being done in a cooperative way.
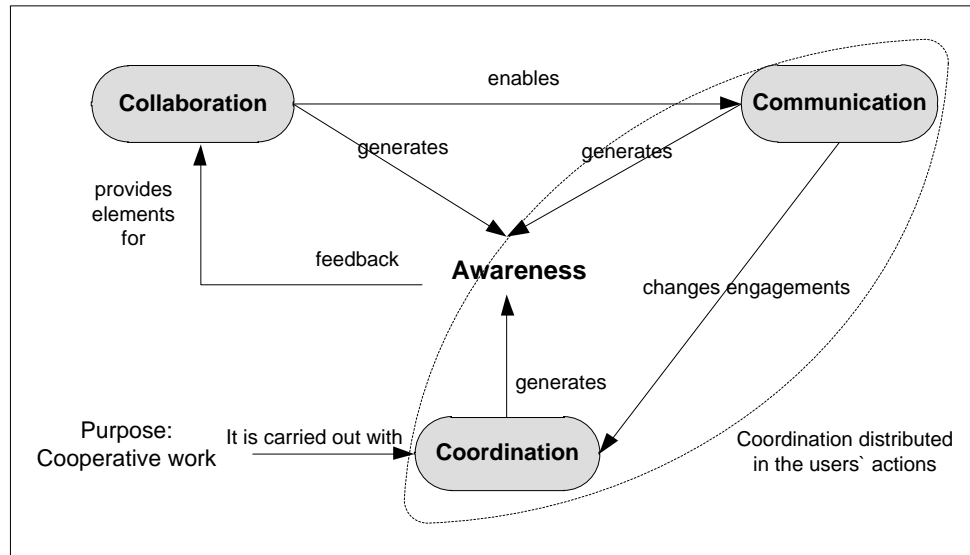
Figure 9 - Diagram of the 3Cs and awareness adapted to cycle 3
Source: produced by the author

This figure presents several input stimuli and one output stimulus. This means that several events of participants that belong to the same group, volunteers or not, must have an awareness element that generates feedback for the collaboration of members of a working group.

The example of Figure 9 highlights that information generated for collaboration and communication does not have to be compulsory once the feedback may not be desired in all moments during the work. On the other hand, the coordination event will always provide some degree of awareness once the workflow could be interrupted and get stuck without information transmission.

5.2 Cycle 2

During cycle 2 of the research-action (not presented in this article) the first artefacts linked to the 3C model emerged and they are reproduced in figures 11 and 12.

Although these two artefacts are discussed collectively with users while the software simulation was taking place in cycle 2, they were not enough to avoid some problems linked to cooperative work that arises from other emergent artefacts that came up during this simulation while the system was in real use. The changes from face-to-face interactions to interactions mediated by the software were not foreseen in an explicit way during simulation of cycle two of action-research.

For example, the user could not always understand which his/her real role was in the system or   suggest alternative ways like parallel registrations while using the functional prototype.

The 3C Cooperation Model applied to the classical requirement analysis

To make an easier understanding of the presentation and interpretation of the obtained data with the application of the proposed model of this paper, Figure 10 reproduces the implemented workflow by the laboratory monitoring software described in 3.3.1 with the several phases the process has.
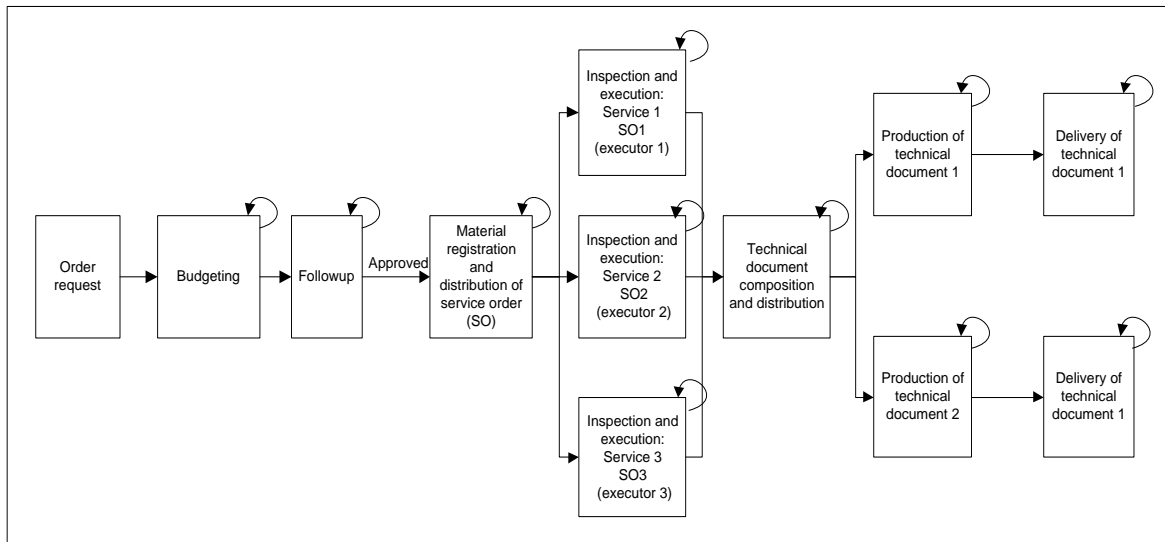
Figure 10 - Final flowchart of the process

The first artefact described in Figure 11 was created due to the user` necessity to have as a starting point of the application a coordination artefact that could show everything that was pendent (each item is listed in Figure 10) to permit the coordination of the execution system of workflow phases in a context where they could be accessed in theory by any user of the group allowing a self-coordinated system, i.e., the actions of the group should lead to its own coordination.

Through this artefact and the representation the user holds in the system, which allows him/her to understand the its total status (awareness), the user organizes his/her activities and defines which shared object of collaboration (in this case, it corresponds to one of the eight phases of the process) will be used through the "Link" column connecting his/her actions with the other users.

Figure 11 - Individual coordination and main page of the software

Whenever there is collaboration mediated by one of these shared objects (see item 5.1), new information is stored in the system allowing communication to happen among the users by changing engagements in the coordination artefacts.

Thus, there is no clear distinction between coordination and communication (see item 4.2) and both are collapsed (Figure 9) in a way that communication always happens in an indirect and asynchronous way as a result of collaboration, which means data inserted into the system.

As there is no explicit object of communication, the coordination is done through an understanding of the relation of the users` actions (Figure 11) and its interrelations with the other users` actions (Figure 12).

Figure 12 - Coordination with group activities

The artefact of Figure 12 (accessed by ""*relatórios*" - reports - of Figure 11) was created because the user of the working group also needed to know what the other participants of the group were doing besides the situation of his/her own work. Thus, they could fit their own activities with the activities of the other group participants. The purpose of this artefact was to show all pendent phases of a particular request, regardless of whom their responsible was. Thus, the user could be aware of how the course of the entire request was.

Figure 12 shows this new coordination artefact that can be seen as: (report of) the request progress. As an example, request 6/06 was used (Figure 12). We can see that this process has some pendant phases that are not with the user "COLLABORATOR NAME 1". For example, the Composition phase of the document is with "COLLABORATOR NAME 4" and one phase of "Document production" is with the user "COLLABORATOR NAME 6".

5.3 Cycle 3

The objective of this cycle is the refinement of requirements attained in cycle 2 through the system in use especially the ones that privilege the cooperative work of the system`s final users.

The 3C model and its relation with the developed research is applied offering to users awareness elements that facilitate the contextualization of individual activities by the comprehension of activities performed by the other group´s participants.

While the collective discussion about new artefacts was taking place, some issues about the dynamics regarding phase transmission within the working group were raised. There was one emergent aspect of this discussion; it was the necessity of visualizing the phase that was with another user in order to perform activities of the user

him/herself, or even the situation where it was necessary to assume another user`s phase to keep going with their own work and the group work (for example, in case of the absence of the responsible for the phase).

In order to meet this need, a solution in which one user could visualize one phase that was not under his/her responsibility was discussed and they could even send it to another user or solve it by themselves (transversal access to the process, which means without its user`s pendency).

Figures 14 and 15 show adopted solutions after this session with the users. The access to it is done through the main page on the tab "*Acesso transversal*" (Transversal access)

Figure 13 - Main page: *acesso transversal* (transversal access)

Figure 14-1 shows filter options and, in this case, one filter per user was inserted. Figure 14-2 shows the backlog for the researched item (in this case, the request number is 606). This backlog aims to allow the user to notice all the pendent phases of that request.

Figure 14 - Transversal access: backlog

Having all pendent phases verified in this list, the user chooses the phase that he/she wants to visualize (Figure 15-1) by clicking the "Confirmar" (Confirm) button (For Figure 15-2 the phase "Composição do documento" (Document composition) was selected and it presents *NOME COLABORADOR 4* (COLLABORATOR NAME 4). Thus, the user will be able to see/edit the content of this phase (see Figure 16) and if

they want, it is also possible to send it to another user (in this case, it will be sent to NOME COLABORADOR 5 (COLLABORATOR NAME *5*), see Figure 16-1)

Figure 15 - Transversal access: choice of phase

Figure 16 - Composition of technical document visualized through the artefact Transversal access

In the discussion of transversal access, the participants of this session guided by awareness elements of Tables 1 and 2 mentioned the following situation: if one user could enter and update a new phase that was not in his/her backlog (under their responsibility), the system should register a background about the access on this phase.

Thus, there was another emergent aspect of the session that was the necessity of an artefact to register who the responsible for the phase was, who accessed it and to whom it was forwarded. The discussion about the characteristics of this artefact should have brought the fact that this same artefact could contribute as a tool to assist in verifying the process flow, it could also contribute to know if there was a sudden closing and to increase the system reliability because users could be aware about who accessed/changed a particular phase.

Figure 17 shows this artefact after its implementation whereas Figure 17-1 presents filter options suggested in the session, and Figure 17-2 highlights an example of a result for the research by the request number, showing the situation described in Figure 16-1 whose phase was with the user "*NOME COLABORADOR 4*" (COLLABORATOR NAME 4) accessed by "*NOME COLABORADOR 1*" (COLLABORATOR NAME 1) and sent to user "*NOME COLABORADOR 5*" (COLLABORATOR NAME 5)

Figure 17 -Background of the request in progress

In data provided by Table 3, we can observe the relation between awareness elements associated to emergent artefacts that came up during this session.

Table 3 - Emergent artefacts of session 2, awareness elements and actions

Artefact/Interview	Awareness elements
Transversal access	What (actions, artefacts and production)
Background of requests in progress	What (background of actions), When (background of events) and Where (background of location)

Source: produced by the author

6. CONCLUSIONS

With the purpose of finding conclusions for this article, we intend to answer the following question related to this action-research:

-        How can the 3C Cooperation Model and awareness be applied to the elicitation of requirements for cooperative work?

This question can be answered by employing the proposed process described in item 4.3 that takes into account the dynamics for the 3C model and awareness application described in 5.1, and it is applicable to systems according to 4.2 with results obtained in field presented in 5.2.

Thus, awareness and the 3C model elements that emerged in cycle 2 of RA (Figures 11 and 12) were not enough for a full accomplishment of awareness in the new computerized environment, in a way that, during cycle 3,the application of new artefacts emerged in the CAW sessions for users through awareness elements described in Tables 1 and 2 with the development of a set of artefacts (Figures 13, 14, 15, 16, and 17):

- Transversal access: the necessity assumed by the group to visualize the phase that was with another user in order to carry on activities of the user him/herself, or even the situation where it was necessary to assume another user`s phase to keep going with their own work and the group work.

- Background of phases of the processes: the necessity of an artefact to register who the responsible for the phase was, who accessed it and to whom it was forwarded. The discussion about characteristics, this artefact should have brought the fact that this same artefact could contribute as a tool to assist in verifying the process flow; it could also contribute to know if there was a sudden closing and to increase the system reliability (subordination).

Thus, by using the process established for cycle 3, it was possible to check that a computerized projected system aiming to meet the cooperative work requirements of an IS must consider face-to-face iteration changes of users in an IS with the purpose of achieving a contact intermediated by the computerized system that presents a less fruitful system to carry out necessary iterations in order to reach the goals of cooperative work.

7. FUTURE WORKING ACTIVITIES

While cycle 2 of this research (not approached in this work) mainly dealt with more transactional requirements of cooperative work where evolution of these requirements was directly checked during the simulation of the future computerized system, in cycle 3, the verification of requirement awareness evolution was not direct in a way that the necessity of developing a new qualitative method to measure these kind of requirements was raised, as Gava (2011a) shows.

Another important aspect is that new interdisciplinary research must be developed in the field of requirements elicitation of cooperative work aiming for quality and fast improvement regarding the discovery of requirements of cycles 2 and 3, as, for example, the use of the concept of instruction for the double suggested by Clot (2000) in cycle 2 and by concepts of explicitation interview in cycle 3 (Vermersch, 2006).

The 3C Cooperation Model applied to the classical requirement analysis

ACKNOWLEDGMENT

REFERENCES

Assis, R. L.(2000). *Facilitando a percepção em ambientes virtuais de aprendizado através da tecnologia groupware*. 148p. Master Thesis - Departamento de Informática, Pontifícia Universidade Católica, Rio de Janeiro, RJ, Brazil.

Boar, B.H. (1985). *Application Prototyping* (1st Edition). New York: John Wiley & Sons.

Brinck, T. & McDaniel, S. E. (1997). Awareness in Collaborative Systems. *Conference on Human Factors in Computing Systems, SIGCHI Bulletin*, 29(4).

Calvão, L.D.,Pimentel, M. & Fuks, H. (2011) Evolução dos Sistemas de Comunicação. *Anais do VIII Simpósio Brasileiro de Sistemas Colaborativos*. Paraty , RJ, Brazil. , pp. 173-176.

Coughlan, P. & Coghlan, D. (2002) Action research for operational management. *Internacional journal of operation & Production management*, 22(2), pp. 220 – 240.

Clot, I. (2000). La formation par l'analyse du travail: pour une troisième voie. In: MAGGI, B. (Ed). *Manières de penser, manières d'agir en éducation et formation*, Paris: PUF.

Daniellou, F. (2007). A ergonomia na condução de projetos de concepção de sistemas de trabalho. In: Falzon, P. (Ed.). *Ergonomia.* São Paulo: Edgard Blücher.

Daniellou, F. & Six, F. (2003). Les ergonomes, les prescripteurs et les prescritions. In: Martin, C.; Baradat, D. (Ed.). *Des pratiques en réflexion - Dix ans de débats sur l'intervention ergonomique.* Toulouse: Octarès Editions.

Dejours, C. (2005). *O fator humano*. (1a ed.). Rio de Janeiro: FGV.

Dourish, P. & Belloti, V. (1992). Awareness and Coordination in Shared Workspaces. *Proceedings of the ACM conference on Computer-supported cooperative work.* Canada.

Ellis, C. A., Gibbs, S. J. & Rein, G. L. (1991). Groupware - Some Issues and Experiences. *Communications of the ACM*, 34(1), pp. 38-58.

Erceau, J., Chaudron, L., Ferber, J. & Bouron, T. (1994). Systèmes personne(s): patrimoines cognitifs et mondes multi-agents, coopération et prises de décision collectives. In: *Systèmes coopératifs: de la modélisation á la conception.* Toulouse: Octarès Editions.

Ferreira, L. L. (1993). Análise coletiva do trabalho. *Revista brasileira de saúde ocupacional.* 21(78), pp.7-19.

Fuks, H. & Assis, R. L. (2001). Facilitating Perception on Virtual Learningware based Environments. *The Journal of Systems and Information Technology*, 5(1).

Fuks, H., Raposo, A. B., Gerosa, M. A., Pimentel, M., Filippo, D. & Lucena, C. J. P. (2007). Inter- e Intra-relações entre Comunicação, Coordenação e Cooperação. *Anais do IV Simpósio Brasileiro de Sistemas Colaborativos* – SBC: Rio de Janeiro - RJ.

Fuks, H., Raposo, A. B., Gerosa, M. A., Pimentel, M., Filippo, D. & Lucena, C. J. P. (2011). Teorias e modelos de colaboração, In: Pimentel, M., Fuks, H. (Org.). *Sistemas colaborativos.* **(**pp. 16 –33). Elsevier.

Gava, V. L. (2011a). *Requisitos de software & cooperação.* (1a ed.) São Paulo: Editora Edgard Blucher.

Gava, V. L., Spinola, M. M., Gonçalves, R. F., Medina, J. M. C. & Tonini, A. C. (2011b). *Collective work characteristics in business information system development.* Revista Gestão da Produção, Operações e Sistemas.

Gava, V. L., Spinola, M. M., Medina, J. M. C. & Tonini, A. C. (2011c). *Proceso para simulación del trabajo cooperativo en la concepción de sistemas informatizados por medio del uso de técnicas de ergonomía del trabajo y cognición.* Revista Espacios (Caracas), (32, pp. 24-30).

Gerosa, M. A. (2006). *Desenvolvimento de Groupware Componentizado com Base no Modelo 3C de Colaboração.* Phd Thesis - Departamento de Informática, Pontifícia Universidade Católica, Rio de Janeiro.

Gerosa, M. A., Fuks, H. & Lucena, C. J. P. Suporte à Percepção em Ambientes de Aprendizagem Colaborativa (2003, novembro). *Revista Brasileira de Informática na Educação*, 11(2).

Gonçalves, R. F., Gava, V. L., Pessôa, M. S. P. & Spinola, M. M. (2005). Uma proposta de processo de produção de aplicações Web. *Revista Produção*, 15(3).

Leffingwell, D. & Widrig, D. (2003) *Managing Software Requirements. A Use Case Approach* (2a ed.). Addison-Wesley: Boston.

Kotonya, G. & Sommerville, I. (1998). *Requirements Engineering (Processes and Techniques)* (1a. ed.) John Wiley & Sons Ltd: England.

Morin, E. (2002). *Ciência com consciência* (6a. ed.). Bertrand Brasil: Rio de Janeiro.

Norman, D. A.(2006). *O design do dia-a-dia.* (18a ed.). Editora Rocco: Rio de Janeiro.

Piaget, J. (1996). *A construção do real na criança.* Ática: São Paulo.

Pinheiro, M. K., Lima, J. V. & Borgwe, M. R. S. (2001). Awareness em Sistemas de Groupware. *Proceedings of IV Jornadas Iberoamericano de Ingeniería de Requisitos y Ambientes de Software*. Santo Domingo: Costa Rica.

Pressman, R.S. *Software Engineering: A Practitioner's Approach* (7a ed.). McGraw-Hill: New York.

Sabetzadeh, M., Finkelstein, A. & Goedicke, M. (2010). Viewpoints. In: Laplante, P. (Ed.). *Encyclopedia of Software Engineering.* New York: Taylor and Francis.

Santos, V.V., Tedesco, P., Salgado & A.C. (2011). Percepção e contexto. In: Pimentel, M., Fuks, H. (Org**.). *Sistemas colaborativos.* (pp. 157 –172). Rio de Janeiro: Elsevier.

Sommerville, I. (2010). *Software Engineering* **(**9a ed.). Pearson Education Limited: Edinburgh.

Souza, G. M. & Castro, J. F. B. (2004). Improving the Separation of Non-Functional Concerns in Requirements Artifacts. *12th IEEE International Requirements Engineering Conference*. Japan.

Thiollent, M. (2011). *Metodologia da pesquisa-ação* (18a ed.). Cortez: São Paulo.

Vermersch, P. (2006). *L' entretien d'explicitation.* ESF: Paris.