

# Activity Theory: a Framework to Software Requirements Elicitation

Luiz Eduardo Galvão Martins<sup>1</sup>, Beatriz Mascia Daltrini<sup>2</sup>

<sup>1</sup>Methodist University of Piracicaba - Unimep  
E-Mail: martins@cce.unimep.br

<sup>2</sup>State University of Campinas - Unicamp  
E-Mail: beatriz@dca.fee.unicamp.br

**Abstract.** In this article we defend the idea that social aspects have strong influence in the software requirements elicitation (Goguen 1993), which drive us to find help in the social sciences. The Activity Theory is a theory developed in the Psychology that focuses the human practices of development process, both the individual and social levels. This theory states that any human action must be understood within a minimal social context, established by an activity. So, we have proposed an approach of software requirements elicitation that have as framework several precepts from the Activity Theory. The case study is developed to show the possibility of using some principles of the Activity Theory in the software requirements elicitation.

**Keywords:** Activity, Activity Theory, Context, Requirement, Requirements Elicitation.

## 1. Introduction.

One fundamental question in the Requirements Engineering is how to find the real user necessities. Researches have proved that many software projects have failed because of the problems in the software requirements elicitation (Boehm 1981; GAO 1992), i.e., the requirements that are got many times are uncompleted, misunderstood and ambiguous.

The correct identification of the software requirements is not an easy task, because the abstract nature of the software. An approach that can be used to a better understanding of the problems found in the requirements elicitation is to divide the problems into two large groups (Brooks 1987; Faulk 1997): accidental problems and essential problems.

The accidental problems emerge because the poor control over the activities developed in the requirements engineering, where we can stand out: low

effort in the requirements elicitation with the user, poor documentation about the requirements, poor revision of the requirements, incorrect specification of the requirements and tendency to initiate prematurely the software development process.

The essential problems are embedded in the requirements elicitation, where we can stand out: difficulty of the user to know exactly what he wants, difficulty of communication between user and developer and the changing nature of the requirements.

The accidental problems can be considered less difficulty to be overcome. The adoption of a systematic process that orient the elicitation, analysis, specification, validation and management of the requirements tend to solve, or at least minimize, the problems of that category.

Nevertheless, the essential problems are more difficulty to be overcome, once they are contained in the requirements nature. The adoption of a systematic process to the requirements engineering, mainly to specification, validation and management of the requirements, will also contribute to overcome the essential problems. However, the problematic that naturally exists in the human comprehension and communication process, which is in the nucleus of the requirements elicitation, will need an approach which take into account the context in which the persons develop their activities and recognize the objects needed to develop them, the historic of evolution of these activities and mediation tools, and others aspects of psychological and social relevancy that affect the users of the software to be developed.

So, we understand that the essential problems of the requirements elicitation will not be able to be solved in an purely technological approach, once the social aspects have strong importance in this activity (Goguen 1993). The most of the software are developed with no one help from the social sciences (like psychology, sociology, anthropology etc.), not approaching in a systematic way the user necessities, both in individual and organization level.

We argue in this article that the utilization of some precepts of the Activity Theory, coming from Soviet psychology, can bring important benefits to the software elicitation process.

## **2. The Activity Theory.**

The Activity Theory can be defined like an interdisciplinary and philosophical framework to study the different ways of human practices of development process, both the individual and social levels. The Activity Theory has three historic roots: the Germany classic philosophy from the 18<sup>th</sup> and 19<sup>th</sup> centuries (from Kant and Hegel); the Marx and Engels's manuscripts, which worked on the concept of activity; and the Soviet psychology, funded by Vygotsky, Leont'ev and Luria. The term "Activity Theory" emerged between 1920 and 1930, in the Soviet Historic-Cultural School of Psychology (Nardi 1996; Kaptelinin 1997).

## **2.1 Basic Principles of the Activity Theory.**

The Activity Theory is formed by a set of principles that constitutes a general conceptual system. The basic principles of the Activity Theory are (Nardi 1996; Kaptelinin 1997):

- *(1) Principles of the unit between activity and consciousness.* It is considered the fundamental principle of the Activity Theory, where activity and consciousness are treated in an integrated way. The consciousness means the human mind like a whole, and activity means the human interaction with its objective reality. This principle states that the human mind emerges and exists like a special component of the human interaction with its environment. The mind is a special organ that appears in the evolution process to help organisms to survive. So, it can be analyzed and understood only within the human activity context.
- *(2) Principle of the object orientation.* This principle focuses on the approach of the Activity Theory for the environment where the human being interacts. Human beings live in an environment that is very important for them. This environment consists of entities that combine all kinds of objective features, including those culturally determined, which influence the ways persons act over those entities.
- *(3) Principles of the hierarchical structure of activity.* The Activity Theory differentiates the human procedures in several levels (activity, action and operation), taking into account the objectives to which these procedures are oriented. The importance of that distinction is determined by the ecological attitude from the Activity Theory. In a real situation, this distinction is frequently necessary to preview the human behavior. So, this distinction is very important to make the differentiation among motives, goals and conditions, that are associated

to activities, actions and operations, respectively.

- (4) *Principle of the internalization-externalization.* This principle describes the basic mechanisms about the mental processes source. It states that mental processes are derived from external actions through the way of the internalization. Internalization is the information absorption process (in several ways) achieved by human mind, which derives from the contact with the environment where the person is located. The externalization is the process contraire to internalization, manifested through acts, in such a way they can be verified and fixed, if necessary.
- (5) *Principle of the mediation.* The human activity is mediated by several tools, both externs (e.g.: an axe, a computer etc.) and interns (e.g.: an heuristic, a concept etc.). The tools are “vehicles” of the social experience and cultural knowledge.
- (6) *Principle of the development.* According to the Activity Theory, to understand a phenomenon means to know how it is developed by itself until its current shape, because it change by the time. Understanding these changes can help to understanding its current state.

Those principles are not isolated ideas, they are closely connected. The nature of the Activity Theory is manifested in that set of principles.

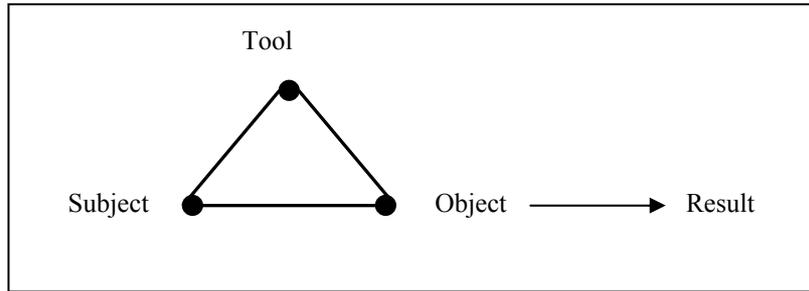
## **2.2. The concept of Activity.**

According to the Activity Theory, one activity is a way in which a subject acts aiming an object. In the individual level, an activity has three elements: subject, object and mediation tool. The subject is the agent who acts upon the object of the activity. The object is the element to which the actions will be directed. An object may be something material, or something at least tangible, as for example, a plan or an idea.

The reciprocal relationship between the subject and the object is always mediated by one or more tools (also called mediation artifacts), that can be instruments, signs, procedures, machines, methods, laws, ways for organization of work etc. Tools always have a role in the mediation process and are used in the process of the object transformation (Nardi 1996).

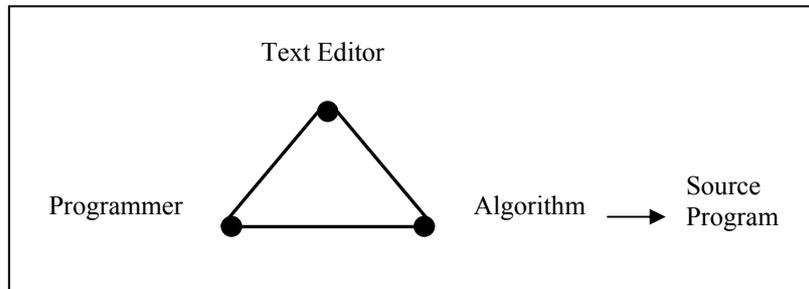
Figure 1 represents the relationship structure, in the individual level,

between the subject and the object in the context of an activity, where the tool assumes an important role in the mediation among them. Through this mediation, some result is obtained. Transforming an object to a result motivates the existence of an activity.



*Figure 1 – Relationship mediated between the subject and the object in the individual level (Nardi 1996).*

In order to exemplify the figure shown above, consider the following activity: “Write a program”. In this case, the subject of the activity would be a programmer; the mediation tool would be a text editor; the object to be transformed would be an algorithm and the result would be the source program ready for compilation (see figure 2).



*Figure 2 – Structure of the activity "Write a program".*

Although the representation of the relationship mediated by the subject and the object in the individual level is useful, this structure is too simple to represent the considerations of the existing systemic relations between the subject and its environment, once these relations are found in a lot of activities.

So, a new element should be added to the structure of the activity: the community. The community is formed by all the subjects which share the same object. When the concept of community is presented, new ways of mediation arise (besides that possibility through the tools). These new ways of mediation are called rules and division of labor (see figure 3).

Rules as a form of mediation between the subject and the community, are implicit or explicit norms established by conventions and social relations in the society. The division of labor is a form of mediation between the community and the object, refers to a form of organizing a community, related to the process of transforming an object into a result.

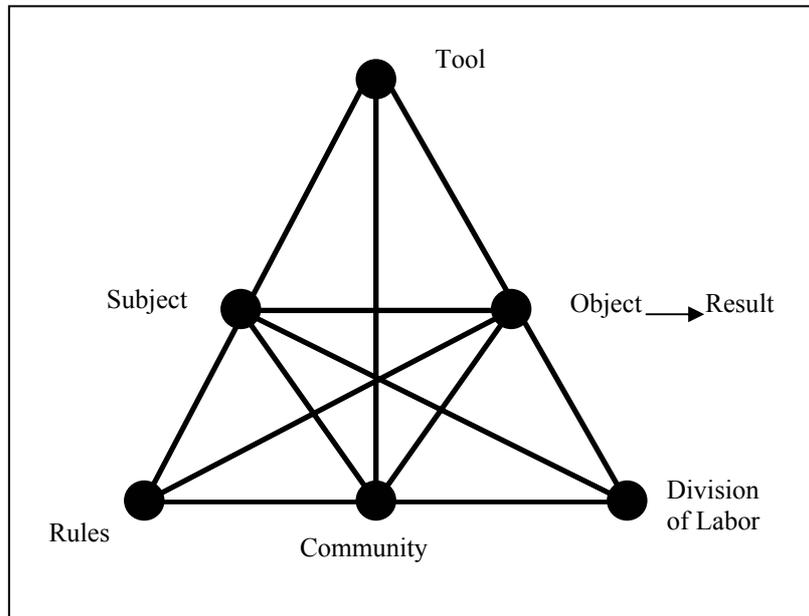


Figure 3 – Systemic Model of an activity (Engeström 1987; Nardi 1996).

All the mediation forms (tools, rules and division of labor) have a historical development of their own, with particular characteristics related to the context in which they were developed.

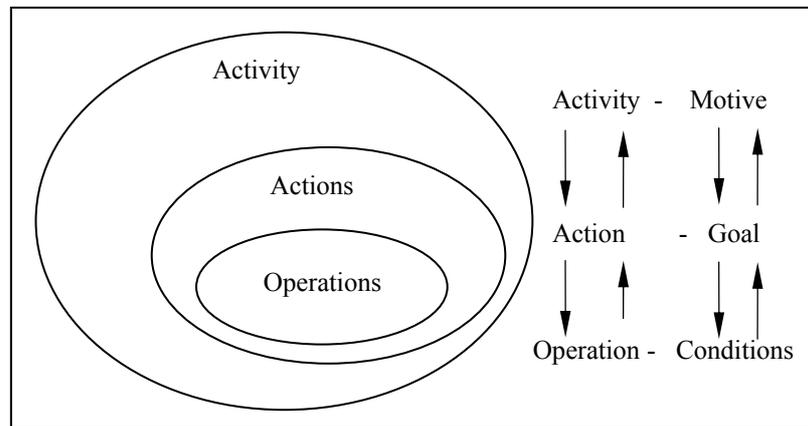
According to the Activity Theory, activities are not static, they evolve, normally in a non-linear way. Each activity has its own history, embedding past “phases”. One historical analysis of its development is often needed to

understanding the current situation.

### 2.3 Levels of an Activity.

An activity is decomposed into actions, and each action is decomposed into operations (see figure 4). Activities are long way formations, their objects are transformed into results not only once, but through a process of several phases or steps.

So, an activity, created in a given moment, passed through a process of evolution, where actions and operations can have been created, eliminated and transformed in order the activity reached its current “format” (Nardi 1996).



*Figure 4 – Hierarchical levels of an activity.*

While an activity is oriented by a motive, the actions are oriented to goals, and the operations oriented to conditions. One activity is performed through cooperative or individual actions and chains or nets of actions that are related one to another for aiming the same goal.

One important characteristic of an action is that it is planned before its effective execution. On the other hand, an operation is executed in an automatic way, without a previous planning. Only an analysis of the current conditions to its execution. The planning of an action is done in a conscious way, using some mental model. The better the model more success the action will reach. This planning to the execution of an action is called orientation.

When an action is performed several times and reach a level of maturity enough to be executed automatically, that is, without a previous planning, then it reaches the level of operation. In this way, one operation was an action which became common in the context of an activity, once is executed with a high degree of repetition inside this context. In table 1 an example of the decomposition of an activity is shown. The activity “Process Sale ” was decomposed into actions “Emit the Sale Bill” and “Emit the Sale Payments”.

<b>Activity</b>	<b>Actions</b>	<b>Operations</b>
Process Sale	Emit the Sale Bill	Fill the fields of the sale bill
		Compute the taxes
		Print the sale bill
	Emit the Sale Payments	Divide the sale into several payment receipts according to their due dates.
		Print the payment receipts

*Table 1 – The activity “Process Sale”*

### **3. An Approach for the Requirements Elicitation from the Activity Theory.**

Some of the psychological approaches use human action as the basic unit of analysis of real life situations. This approach can offer good results as the action in question is analyzed in a isolated way, e.g.: situations designed to laboratory experiments.

However, in the real life, the human action should be analyzed into a context, so it makes sense and it is understood. According to the Activity Theory, a minimum context is given when human action is analyzed inside an activity (according to the concept of activity mentioned in section 2.2). So, the concept of activity is then presented as the basic unit of analysis of situations.

In order to get the software requirements in an adequate way, it is necessary to understand, among other things, the activities performed by the agents involved in the system which will implement the future software.

### 3.1 Showing a Case Study.

The case study presented is about the construction of a software to control the protocols of a secretary's office of an university. The following problem declaration was obtained from a real situation: an open interview (Goguen 1993) was performed with the secretary.

#### Initial Declaration of the Problem

"The system of protocols consists in controlling documents in and out of the secretary's office. Considering any document that goes in and out of the office, it is generated a number for the protocol and the fields from the registration board of protocols are filled (figure 5).

(1) Protocol n °	(4)From	(5)Date	(6)Rec	(7)To	(8)Date	(9)Rec
(2)Name:						
(3)Area:						

*Figure 5 - Registration Board of Protocols.*

The fields presented in the registration board of protocols mean:

1. Number of protocol
2. Name of the person who submitted the protocol (local/department)
3. Area of Interest
4. From (local source/ person who signed)
5. Arriving date
6. Person who received (name of the person in charge)
7. Destination (the person to whom the protocol is addressed )
8. Leaving Date (date of the submission of the protocol to the receiver)

9. Receiver Signature (or some kind of information that can identify that the receiver got the protocol)

Those information are annotated in a way to maintain registrations about the documents which circulate by the secretary's office.

Nowadays, this type of control is done without the use of computers. Protocols are registered in a book, where each page can have until 10 registers of protocols. All pages have a number.

### **3.2 The Approach for the Requirements Elicitation.**

The approach that has been adopted for the requirements elicitation of the case study presented consists of the following steps:

1. Identify procedures performed in the system which can be classified as activities.
2. Identify for each activity: subject, tool, object, community, rules, division of labor and results (representation of the systemic model of activity).
3. From the systemic model of activity, decompose the activities into actions and operations.

#### 1. Procedures Classified as Activities

In order to identify procedures (or processes) that can be classified as activities the principles (1), (2) and (3) of the Activity Theory (mentioned in section 2.1) will be utilized. Thus, can be classified as activities, for example, the following procedures:

- Create a protocol
- Update a protocol
- Consult a protocol by its date

#### 2. Systemic Models

After the identification of the activities, the systemic models of the activities can be developed. Through the systemic models, the elements that compose the activities "create a protocol", "update a protocol" and "consult a protocol by its date" should be obtained, according to the figures 6, 7 and 8 respectively. Those models take into account the principles (2) and (5) from

the Activity Theory.

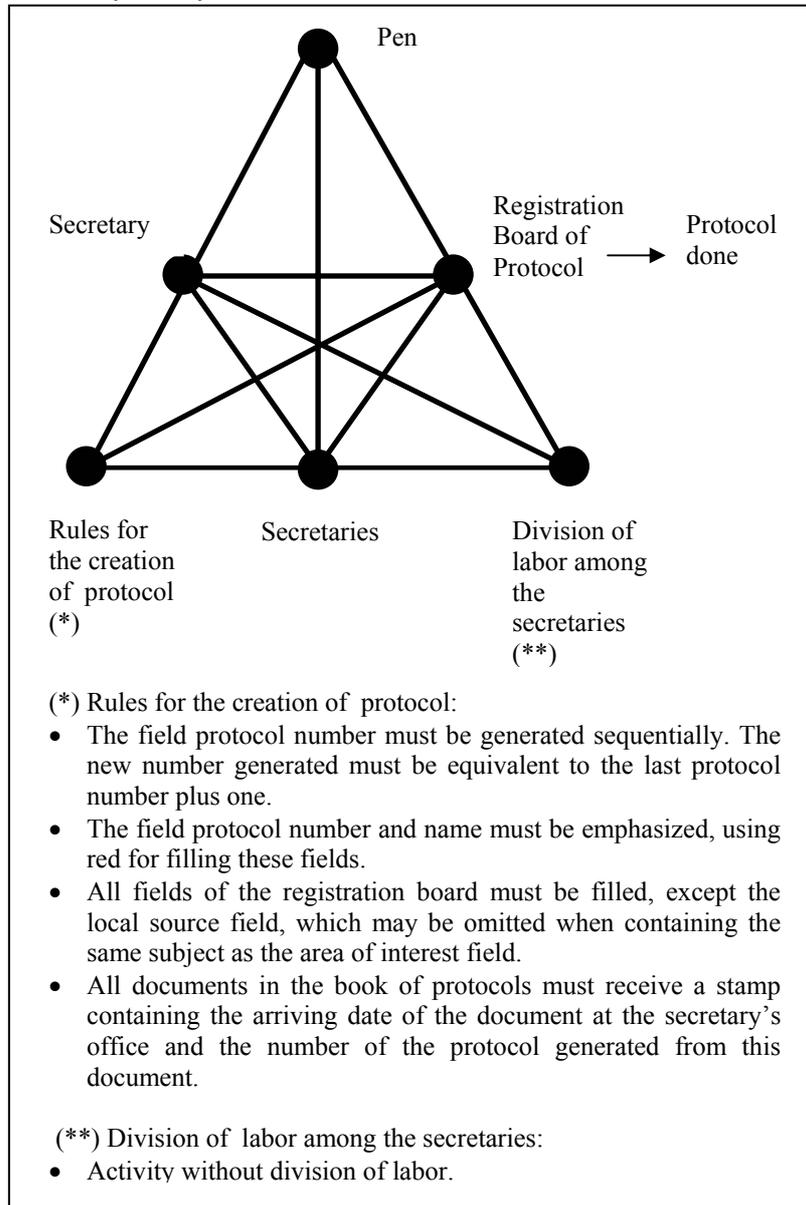


Figure 6 – Systemic Model of the Activity "Create Protocol".

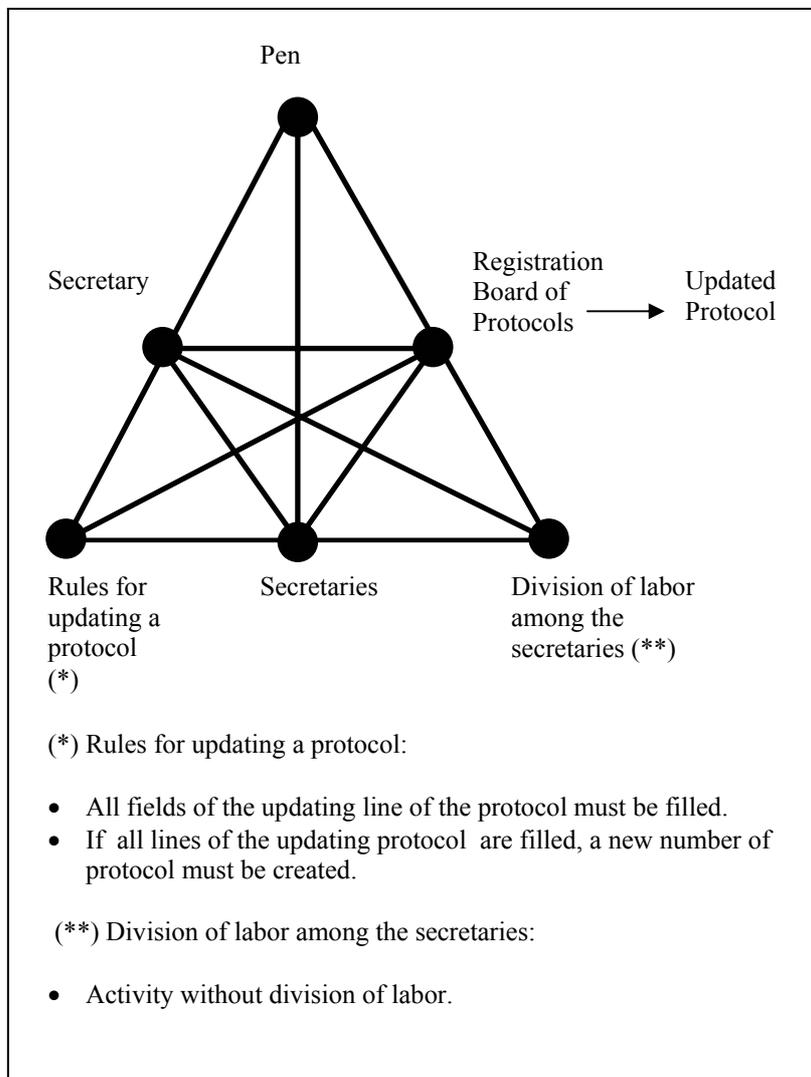


Figure 7 – Systemic Model of the activity "Update Protocol".

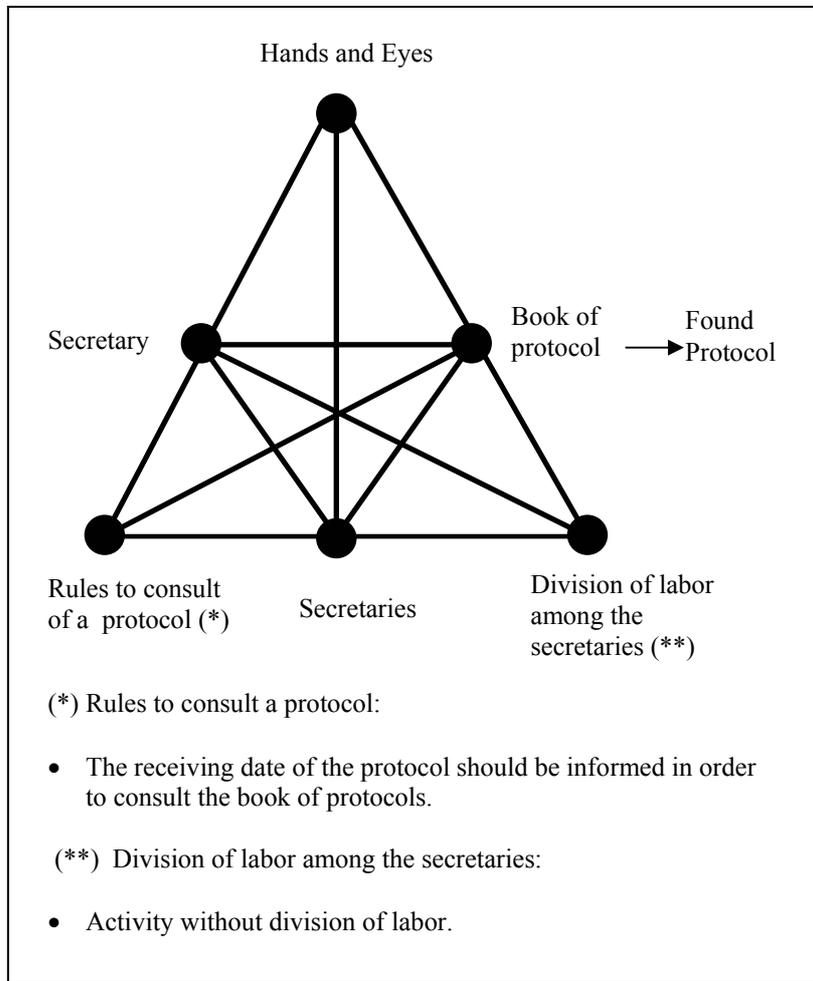


Figure 8 - Systemic Model of the activity "Consult protocol by its date".

### 3. Decomposition of an activity into Actions and Operations.

In order to represent in a more detailed way the activities mentioned in the previous section, they will be decomposed into actions and operations, according to the principle (3) from the Activity Theory.

The decomposition of activities “Create Protocol”, “Update Protocol” and “Consult Protocol by its Date” is presented through the tables 2,3 and 4.

<b>Activity</b>	<b>Actions</b>	<b>Operations</b>
Create protocol	Generate protocol number	Verify last protocol number
		Add one to the last protocol number
		Fill field “protocol number”(1)
	Fill first line of the registration board of protocol	Fill field (2)
		Fill field (3)
		Fill field (4)
		Fill field (5)
		Fill field (6)
		Fill field (7)
		Fill field (8)
		Fill field (9)
	Submit document	Stamp document to be sent
		Copy protocol number in the stamped document
		Send document to receiver

*Table 2- Decomposition of the activity "Create protocol".*

<b>Activity</b>	<b>Actions</b>	<b>Operations</b>
Update protocol	Find registration board of protocol in the book of protocols	Verify protocol number in the received document
		Find the correspondent number in the book of protocols
	Fill next line in the registration board of protocols	Fill field (4)
		Fill field (5)
		Fill field (6)
		Fill field (7)
		Fill field (8)
		Fill field (9)

*Table 3- Decomposition of the activity "Update Protocol".*

Activity	Actions	Operations
Consult protocol by date	Find protocols	Specify date to consult
		Find protocols numbers according to the specified date
		Inform consulted protocols

Table 4 - Decomposition of the activity "Consult protocol by date".

#### 4. Conclusions.

It seems that the systemic relations existed in an activity context contribute to a more careful requirements elicitation, once it takes the person performing the elicitation into considering important elements which are necessary for the understanding of a problem. Such elements are: subject, mediation tools, object, community, rules and division of labor.

The hierarchical structure of an activity, composed by actions and operations and their "movements" along the historical development of an activity also contribute to a better understanding of the analyzed problem.

Nowadays there are some research being developed in the requirements elicitation area, which uses the concept of scenario<sup>1</sup> in order to support the requirements elicitation of the macro-system (Breitman 1998; Leite 1997).

Considering that the concept of scenario can be structured through concepts like context, episode, objective, actor and resource (Breitman 1998), a parallel can be presented between the concepts of scenario and activity. This parallel shows that a lot of their elements are equivalent, for example: an actor in the scenario concept is a subject in the activity concept; a resource in the scenario concept is an object in the activity concept; context in the scenario concept is determined by the rules of the activity, the episode in the scenario concept is an action in the activity concept and the goal of the scenario is the result of an activity.

Nevertheless the activity concept brings more elements inside its structure than the scenario concept: community, mediation tools, division of labor inside the community, and operations. So, we believe that several precepts from the Activity Theory can contribute for process of requirements

---

<sup>1</sup> The concept of scenario is used in several objected-oriented analyses method (Rumbaugh 1991; Jacobson 1992; Booch 1994).

elicitation based on scenarios.

We argued in the introduction section that we can divide the problems faced in the requirements elicitation of the software into two major groups: accidental problems and essential problems. The essential problems contain the real difficulties in the requirements elicitation process. We believe that some of the precepts of the Activity Theory can contribute to the overcoming some of those difficulties.

## References

- Boehm, B., *Software Engineering Economics*, Prentice-Hall, 1981.
- Booch, G., *Object-Oriented Analysis and Design with Application*, Benjamin/Cummings, 1994.
- Breitman, K. K. e Leite, J. C. S. P., "Suporte Automatizado à Gerência da Evolução de Cenários", Workshop de Engenharia de Requisitos - WER98, 1998.
- Brooks, F., "No Silver Bullet: Essence and Accidents of Software Engineering", *Computer*, Apr. 1987, pp. 10-19.
- Engeström, Y., "Learning by Expanding", Helsinki: Orienta-Konsultit, 1987.
- Faulk, S. R., "Software Requirements: A Tutorial", *Software Requirements Engineering*, 2<sup>nd</sup>. Ed., IEEE CS Press, 1997, pp 128-149.
- GAO - US General Accounting Office, *Mission Critical Systems: Defense Attempting to Address Major Software Challenges*, GAO/IMTEC-93-13, December 1992.
- Goguen, J. A. and C. Linde, "Techniques for Requirements Elicitation", *Software Requirements Engineering*, 2<sup>nd</sup>. Ed., IEEE CS Press, 1997, pp 110-122.
- Jacobson, I. *et al.*, *Object-Oriented Software Engineering*, Addison-Wesley, 1992.
- Kaptelinin, V. and Nardi, B. A., "Activity Theory: Basic Concepts and Applications", CHI 97 Electronics Publications: Tutorials, march/1997, <http://www.cwi.nl/~steven/chi97/proceedings/tutorial/bn.html>.
- Leite, J.C.S.P. *et al.*, "Enhancing a Requirements Baseline with Scenarios", *Requirements Engineering Journal*: Springer-Verlag London Limited Vol. 2, N. 4 pp 184-198, 1997.
- Nardi, B. A., *Context and Consciousness - Activity Theory and Human-Computer Interaction*, MIT Press, 1996.
- Rumbaugh, J. *et al.*, *Object-Oriented Modeling and Design*, Prentice Hall, 1991.