

An Approach of Software Requirements Elicitation Based on Activity Theory

Luiz Eduardo Galvão Martins
(*Graduate Student*)
Methodist University of Piracicaba
lgmartin@unimep.br

Beatriz Mascia Daltrini
(*Supervisor*)
State University of Campinas
beatriz@dca.fee.unicamp.br

Abstract

The goal of this work is to show an approach of software requirements elicitation based on Activity Theory. This theory has been developed in the psychology area and takes into account several important elements to analyze the context where people perform their actions. Considering the activity as unit of analysis the Activity Theory offers an interesting framework to structuring and organizing the actions performed by the system stakeholders. We intend to use this framework to help in the discovery and capture of software requirements.

Keywords

Activity, Activity Theory, Software Requirements, Software Requirements Elicitation

1. Introduction

A classical problem in Requirements Engineering is how to determine real user needs and transform them in safe software requirements. Researches have proved that many software projects have failed because of the problems in Software Requirements Engineering [Boe81] [Gao92]. Specifically, the requirements that are obtained are often incomplete, misunderstood, and ambiguous.

The discovery of software requirements is not an easy task because of the abstract nature of software. To discover software requirements is a task performed in the elicitation process, the first step of Requirements Engineering [Kot98]. The problems found in requirements elicitation can be divided into two categories [Bro87] [Fau97]: accidental problems and essential problems. The accidental problems emerge because of poor control over the activities developed in requirements engineering: 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 requirements elicitation are: difficulty of the user to know exactly what he wants, difficulty concerning the communication between user and developer, and the changing nature of the requirements.

The accidental problems can be considered less difficult to 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 requirements engineering, mainly to specification, validation and management of the requirements, will also aid in overcoming the essential problems. However, the problems that naturally exist in the human comprehension and communication process, which is in the nucleus of the requirements elicitation, will need an approach which takes 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 relevance that affect the users of the software to be developed.

Thus the essential problems of requirements elicitation will not be solved in a purely technological approach when social aspects have strong importance in the activity [Gog93]. The majority of software is developed with no one help from the social sciences (such as psychology, sociology, anthropology etc.), not approaching in a systematic way the users' necessities, both at individual and organizational levels. We argue in this article that the utilization of some precepts from Activity Theory, coming from Soviet psychology, can bring important benefits to the software elicitation process.

2. Related Works

In [McG00] is presented an approach to modeling “softer” aspects of the software development process. This approach considers that several notions from Activity Theory can offer useful support for the critical “people-related” applications. Such notions include intentionality, history, mediation, motivation, understanding, communication, culture and context.

There are some research methods being developed in the requirements elicitation area that use the concept of scenario to support requirements elicitation of the macro-system [Bre98]. Considering that the concept of scenario can be structured through concepts like context, episode, objective, actor, and resource [Bre98], a comparison can be presented between the concepts of scenario and activity.

Experiment results are presented in [Taw98] where textual scenario analysis were used in the requirements elicitation process. Such process is called CREWS¹-L’Ecritoire. This approach proposes to exploit a bi-directional coupling between goals and scenarios to support the requirements elicitation process.

3. Methodology

The methodology of work that has been adopted consisted of:

- Study of software development methodologies;
- Study of Activity Theory;
- Study of Requirements Engineering;
- Development of a case study;
- Preparation and submission of papers to conferences;
- Comparison between activity and use case concepts;
- Planing of the thesis.

Study of software development methodologies.

Several software development methodologies were studied, among them OOAD [Boo94], Fusion [Col94], OMT [Rum91], OOSE [Jac92] and USDP [Jac98]. All these methods offer a lot of modeling techniques to requirements specification, but few attention has been spent to requirements elicitation.

Study of Activity Theory.

The study of the Activity Theory has been focused upon the concept that an activity offers a context to understand the actions performed by the subjects involved in the activity concept [Kuu96]. According to the Activity Theory this context is formed by elements like subject, mediation tool, object, rules, community and division of labor. We believe that the capture of these notions during the elicitation process can help to discovering software requirements.

¹ Cooperative Requirements Engineering With Scenarios

Study of Requirements Engineering.

The study of Requirements Engineering has been focused upon the requirements analysis and elicitation [Kot98] [Fau97], which are very tied activities.

Development of a case study.

A case study was developed to apply some precepts from Activity Theory to the requirements elicitation process. The case studied involved the construction of a software to control the protocols of a secretary's office of an university. 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.

The approach of requirements elicitation used in the case study consisted of three steps [Mar99]: 1) To identify procedures performed in the system, which can be classified as activities; 2) To identify for each activity: subject, tool, object, community, rules, division of labor and results (the relationship of these elements represents the systemic model of activity); and 3) From the systemic model of activity, to decompose the activities into actions and operations.

Preparation and submission of papers to conferences.

After the case study, we prepared and sent papers to some conferences. The goal of this initiative was to show the potential of usage of Activity Theory in the requirements elicitation process. This idea was presented at *14th IEEE International Conference on Automated Software Engineering* (ASE'99), *13th Brazilian Symposium on Software Engineering* (SBES'99) and *II Workshop on Requirements Engineering* (WER'99).

Comparison between activity and use case concepts.

A comparative study between activity and use case concepts was developed, searching by similarities and differences. Several elements present in the concept of activity have not similar in the concept of use case, such as community, division of labor and mediation tools.

Planing of the thesis.

Finally, we are planing the thesis and developing other case studies to refine the software requirements elicitation method based on Activity Theory.

4. Applicability

This work intends to purpose a method to software requirements elicitation taking into account that the context analysis is very important to help discover real user needs. We believe that an activity offers a good unit of analysis to understand the actions performed by the actors evolved in the system, as defended by Activity Theory.

The applicability of the approach purposed is directed to systems whose human factors, such as communication and comprehension (essential problems in requirements elicitation), assume a very important role in the requirements definition.

Nowadays there are some research methods being developed in the requirements elicitation area that use the concept of use case and scenarios to support requirements elicitation of the macro-system [Bre98]. We believe that some precepts from Activity Theory can be applied in use case and scenarios approaches.

We argued in the introduction 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 Activity Theory can contribute to overcoming some of those difficulties.

5. Conclusion

The article presented tried to offer an overview about the work performed until now and the relevance and applicability of Activity Theory to the software requirements elicitation process. We defend that the systemic relations that exist in an activity context and the notion of unit of analysis through the activity contribute to more careful requirements elicitation. However, it is essential for the person performing the elicitation to consider important elements necessary to the understanding of a problem. Such elements include 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.

At the moment we are refining the approach of requirements elicitation used in the case study. The initial approach was divided into three steps (as explained in chapter 3), which are being expanded in the following steps: 1) To identify relevant activities in the context of the system; 2) To identify the results and motives of the activities; 3) To identify the triad: subject, mediation tool and object; 4) To identify the relevant social aspects; 5) To describe the social rules involved in the context; 6) To model the activities through Engeström’ diagram (systemic model); 7) To decompose the activities into actions and operations; and 8) To describe the goals of the actions.

References

- [Boe81] Boehm, B., *Software Engineering Economics*, Prentice-Hall, 1981.
- [Boo94] Booch, G., “Object-Oriented Analysis and Design with Application”, Benjamin/Cummings, 1994.
- [Bre98] Breitman, K. K. e Leite, J. C. S. P., "Suporte Automatizado à Gerência da Evolução de Cenários", Workshop de Engenharia de Requisitos - WER98, 1998.

- [Bro87] Brooks, F., "No Silver Bullet: Essence and Accidents of Software Engineering", *Computer*, Apr. 1987, pp. 10-19.
- [Col94] Coleman, D. *et al.*, "Object-Oriented Development: The Fusion Method", Prentice Hall, 1994.
- [Fau97] Faulk, S. R., "Software Requirements: A Tutorial", in *Software Requirements Engineering*, 2nd. Ed., IEEE CS Press, 1997, pp 128-149.
- [Gao92] US General Accounting Office, *Mission Critical Systems: Defense Attempting to address Major Software Challenges*, GAO/IMTEC-93-13, December 1992.
- [Gog93] Goguen, J. A. and C. Linde, "Techniques for Requirements Elicitation" in *Software Requirements Engineering*, 2nd. Ed., IEEE CS Press, 1997, pp 110-122.
- [Jac92] Jacobson, I. *et al.*, "Object-Oriented Software Engineering", Addison-Wesley, 1992.
- [Jac98] Jacobson, I., Booch, G. and Rumbaugh, J., "The Unified Software Development", Object Technology Series, Addison-Wesley, 1998.
- [Kot98] Kotonya, G. and Sommerville, I., "Requirements Engineering: Processes and Techniques", John Wiley and Sons, 1998.
- [Kuu96] Kuuti, K., "Activity Theory as a Potential Framework for Human-Computer Interaction" in *Context and Consciousness - Activity Theory and Human-Computer Interaction*, MIT Press, 1996, pp. 17-44.
- [Mar99] Martins, L. E.G. and Daltrini, B. M. "An Approach to Software Requirements Elicitation Using the Precepts from Activity Theory". Proceedings of the 14th IEEE International Conference on Automated Software Engineering, Florida, 1999.
- [McG00] McGrath, M. G. and Uden, L., "Modelling 'Softer' Aspects of the Software Development Process: An Activity Theory Based Approach", Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000 IEEE.
- [Rum91] Rumbaugh, J. *et al.*, "Object-Oriented Modeling and Design", Prentice Hall, 1991.
- [Taw98] Tawbi, M. *et al.*, "Guiding the Process of Requirements Elicitation through Scenario Analysis: Results of an Empirical Study", Proceedings of the 10th International Workshop on Database & Expert Systems Applications, 1998 IEEE.