

# Elicitation Strategies for Web Application Using Activity Theory

**Nik Azlina Nik Ahmad<sup>1\*</sup>, Zarinah Mohd. Kasirun<sup>2</sup>**

(1) *Software Engineering Section, Malaysia Institute of Information Technology, University Kuala Lumpur, Malaysia*

(2) *Department of Software Engineering, University of Malaya, Kuala Lumpur, Malaysia*  
nikazlina@miit.unikl.edu.my; zarin@fsktm.um.edu.my

Received: 2011/06/29 ; Accepted: 2011/07/07 Pages: 1-13

---

## Abstract

*Requirements engineering (RE) is often seen as an essential facet in software development. It is a vital process before each project starts. In the context of systems engineering, an understanding and application of systems theory and practice is also relevant to RE. The contexts in which RE takes place habitually involve human activities. Therefore, RE needs to be sensitive to how people perceive and understand the world around them, how they interact, and how the sociology of the workplace affects their actions. This paper will propose a new approach in requirements elicitation process by considering the human intervention in each activity involve with the development process. Web application development requires proper elicitation process because it has its own distinctive features. Without proper elicitation process, the requirements will turn to be inaccurate and incomplete thus will increase the ambiguity. Therefore, this paper will examine the role of the activity theory in order to decrease the impact of the said problems or to avoid them. Some analysis will be made over the theory in order to discover its sustainability in promoting and maintaining the quality of the requirements, with the aim to underline better way of eliciting the requirements for the Web application.*

**Keywords:** *Activity Theory, Requirements Engineering, Requirements Elicitation*

---

## 1. Introduction

Requirements capturing is a vital activity for every software development project. However, it is also the crucial problem that needs to be solved ever since the Web application development was first introduced [1]. On the other hand, most of the methodological solutions that were proposed for Web applications development only focus on the conceptual models but do not pay much attention to the requirements engineering activity [2]. This is supported by [3,4,5], which stated that new requirements engineering methods for Web applications are needed because the existing methods are mainly focused on the traditional software, which is not appropriate to support the distinctive features of Web application development. Since Web is high interactivity application, thus it should support the (a) navigational (b) organizational (c) complexity and (d) accessibility issues [6,7].

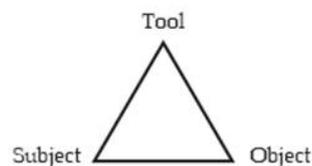
There is always a decision making process involve in every software development and the hardest part is to make a correct decision at all time. Deciding on ‘what to build’ requires thorough understanding on the requirements engineering activities. While the most difficult part of the requirements elicitation process is obtaining a complete and

consistent set of requirements [8]. Therefore requirements elicitation, the first phase of the software development process, is the most critical process of requirements engineering and possibly of the entire software process [9]. Having the idea of the importance of the navigational, organizational, complexity, as well as accessibility in the Web application, thus the involvement of the human activity should be considered during the requirements gathering. Activity theory strives on understanding a situation by engaging various aspects of systemic and individual oddity, which make the analysis of organizational situations richer and deeper [7]. Having its basic unit known as ‘activity’, plus its focus on the human interaction with its relevant environmental context [10], we believe that applying activity theory to requirements elicitation will help the practitioners to analyze the problems better which possibly will overcome the issues listed above. Therefore, this paper proposes an improved process for software requirements elicitation for Web application development by considering the human activity.

## 2. Related Work

Software requirement has been defined by IEEE as a condition or capability that must be met or fulfilled by a system to satisfy a contract, standard, specification, or other formally imposed documents [11]. In the meanwhile, requirements engineering is a necessary process involved during the software development with the purpose to identify, document, and validate the requirements [12,13]. Requirements elicitation is the first phase in requirements engineering which make it vital. Mistakes made in elicitation have been shown many times to be major causes of missing the customers’ real needs which might lead to a systems failure [14,15,16] which later turn to be very costly to either build new software or to correct the mistakes [7,17]. Additionally, according to a survey [18], Web application developers claim that most of their Web development projects are running over budget and time because of problems in capturing requirements and poor communication between themselves and their clients and such problems may cause project failure [16]. Thus, requirements elicitation is an important issue which cannot be left unsolved.

Since it was first introduced, activity theory has been explored by many researchers on its relevancy with the software development. Actions performed by human always reflect to certain mission which occurs via various tools [19]. He then introduced a triangle model, shown in Figure 1, which involves only subject, object with a tool as a mediator.



*Figure 1. Activity Theory*

This work has later been expanded. Having analysis on the subject alone is not sufficient [20]. He then proposed a hierarchical level of activity which results in better

activity analysis for each problem addressed (as shown in Figure 4). A later work [21] proposed a new idea of having more elements in the activity model, considering the community, rules and division of labor as part of the activity process (Figure 2). He saw the structure as one cohesive unit which reflects to each other. This idea resulted to better analysis of problems; however, his work is a kind of general as Engeström did not relate his work with any software development process.

Better explanations were given for activity theory [22], as they highlighted the better way of understanding human activities for collaborative activities. After that, [23] started looking at the potential use of the activity theory in requirements elicitation process by arguing the idea of the accidental and essential problems from the Soviet psychology as they believed that systemic relations which exist in the concept of activity can support better requirements elicitation process which also agreed by [7], whom viewed activity theory as a mechanism that makes the problem analysis of organizational situations more affluent. In contrast, some issues in Web application requirements such as complexity and accessibility have been outlined by [4] through a case study. However their work focused only on how to capture the Web requirements other than finding solutions for those issues and the focal point is not an activity theory. A later work [6] focused more on understanding the social and cultural aspects of users in web development using an activity theory. Other issues such as organizational and navigational were described. This work stressed on how to relate the sociology aspects for the Web application development, however, they only focused on those two issues and neglected about the other issues suggested in [4].

Therefore, this paper will use the activity theory to solve the remaining issues of Web application requirements in the effort to improve the requirements elicitation process.

### **3. Challenges in Requirements Elicitation**

Requirements elicitation is the process of discovering the requirements of a software project to determine the needs of stakeholders [24] and it is the first activity to happen in requirements engineering process [12] before we can further move to analyze, specify and validate the users' requirements [13]. Several software requirements elicitation techniques have been used by developers, to assure that the requirements are complete, consistent and correct. However selecting a technique for elicitation is often difficult since many techniques available, but not all of these techniques are suitable for every type of software project [24]. In addition, the current requirements engineering models could not provide a theoretical basis for understanding 'regularly patterned' human activity [25] which is very important for the web application development.

Before any other software development activities take place, the elicitation process must first be properly completed to avoid any unwanted changes and risks which may affect the project's resources. Since requirements elicitation is one of the most difficult knowledge-intensive processes to model [26], therefore the improper elicitation practice may lead to various problems and issues. As reported in [15,27,28], the requirements elicitation process is challenged by different factors as shown in the following Table 1.

**Table 1. The Challenge Factors in Requirements Elicitation**

The Challenge Factors in Requirements Elicitation	
Difficulties	Description
Incomplete	- Incomplete understanding of needs - Incomplete domain knowledge
Incorrect	- Ill-defined system boundary - Misunderstanding system purpose
Ambiguous	- Synonymous and homonymous terms - Untestable terms
Inconsistent	- Conflict /different views of different users - Omitting 'obvious' information
Unfixed	- Fluctuating requirements - Continuous acceptance of additional requirements
Excessive	- Unorganized bulky information source - Too many requesters - Over (unnecessary) commitment

### 3.1 Issues in Web Application

Thinking of a Web application development requires us to consider its distinctive features. This means, requires better elicitation process of the requirements as well. As highlighted in Section 1, several issues that need to be considered during the Web application development are (a) navigational, (b) organizational, (c) complexity, and (d) accessibility. A solution for both (a) and (b) have been given by [6]. In brief, they opined that *navigational* structure should allow the users to feel at ease in navigating through the information in a non-linear way. With lots of information being placed over the Web, it is crucial to properly present the information with the use of appropriate links and nodes. To avoid navigational problems, the navigational structure must be depicted before the development take place. In the meantime, the *organizational* issues must be viewed in social context rather than technical. It needs to be sensitive on how people understand and interact with their surroundings and how the surroundings affect their actions. While the remaining (c) and (d), as suggested by [7] requires further amplification and will be discussed in this paper.

#### (c) Complexity

The Web applications are becoming more complex each day. This includes the content, information changes, availability, and performance. During it first introduced, Web pages were used to host only text and images. However, the use of Web pages has become more complex since it includes various content types [1], ranging from videos to scripts as well as rich Medias. The use and arrangement of those contents will affect the users' reactions. Thus, the requirements elicitor plays an important role in depicting the essential and useful contents from the stakeholders to avoid the complexity. The solution of this complexity issues will be discussed in Section 6.

#### (d) Accessibility

Web accessibility is about people being able to get, interact and use Web content according to their needs and preferences. It concerns about satisfying more people in more situations and should be understandable too. Therefore, the requirements should be made available to the designers in less complicated and simple way; otherwise, it

will turn the design to be wrong or cluttered. Hence, the requirements should be made easier to understand, to avoid any misleading or misunderstanding of the requirements. If this issue cannot be overcome, the misinterpreted requirements may lead to a wrong system design and will not benefit the users. In order to find a way to reduce this difficulty, a solution will be provided in Section 6.

#### 4. Activity Theory

As explained in Section 2, Vygotsky's notions of mediated social processes known as activity theory was developed [20]. This theory distinguishes between actions, operations, and activity. The basic unit of analysis in activity theory is human activity; in which the theory focuses on the interaction of human activity and consciousness within its relevant environmental context [29]. It is fairly described that human activities are aggravated by certain needs in order for the people to achieve certain purpose. Thus it makes the activity independent which means, it cannot exist as an isolated entity. The idea of activity theory is focusing on the notion of 'activity' rather than on individuals or collectives, where the activity focuses on certain aspects of human actions [7]. This approach encounters the human action with its correlated elements. The basic idea of the activity is having a *subject* (act as an agent) to perform certain activity using *tool(s)* to achieve an *object* (objective or purpose) which later will transform an object into outcome(s) [30]. The main thing that distinguishes activity systems from one another is the difference in their *objects* and the relationship between those elements is mediated, but not directed. The basic structure of an activity can be illustrated as in Figure 2.

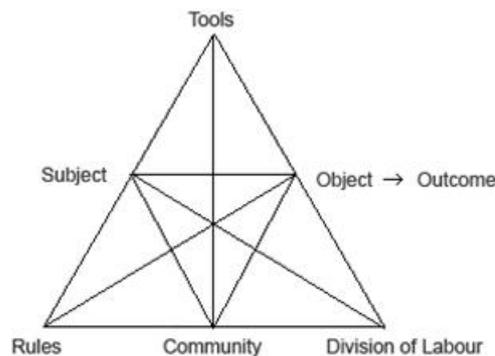


Figure 2. The Basic Structure of Activity Theory

The activity theory focuses on what the agent is actually do within a community and [7] suggested that software development projects are better and effectively to be analyzed as 'activities'.

Table 2. Explanation on the Structure of Activity Theory

Struc-ture	Explanation
Tool	An instrument or artifact used by the <i>subject</i> to transform the <i>object</i> . (i.e. computers, operating systems, software, instruments, signs, UML, procedures, machines, Internet, methods, techniques, documents, laws, books, paper, pen, etc)
Subject	Agent or human actor whose point of view / activity is analyzed (user of specific task – customer, admin, clerk, nurse, lecturer, student, etc)
Object	Object to be transformed by the <i>subject</i> using specific <i>tool(s)</i> (materials or abstract entity)
Result	The final product gained from the transformation process between <i>subject</i> and <i>object</i> , mediated by <i>tool(s)</i> (the outcome)
Rules	Set of regulations and procedures to be followed by the community during the implementation of the activity (i.e. standards, organizational practices and policies, regulations, etc)
Community	Is formed by all the <i>subjects</i> which share the same <i>object</i> (group of <i>subjects</i> ) <i>Table 2 Continued</i>
Division of Labor	Relationship between <i>object</i> and <i>community</i> is mediated by the <i>division of labour</i> . How the activity is distributed among the members of the community or the role each individual in the community plays in the activity (i.e. manager, administrator, web developer, clerks, etc).

By considering each structure during the requirements elicitation, the outcome will be more precise and complete since the requirements is gathered using specific structure, where the practitioner needs to fulfill the structure in order to accomplish the requirements gathering. The practitioner now has a guide rather than just asking various questions to the end user. The requirements will also be more organized as the relationship among the structures is determined earlier. Thus, this approach can offer good results as the activities are not considered in isolated, but closely connected to the subject, mediating tools or technologies and its environment.

A better notation of each activity is represented in the following Figure 3 to demonstrate individual entity's relationship.

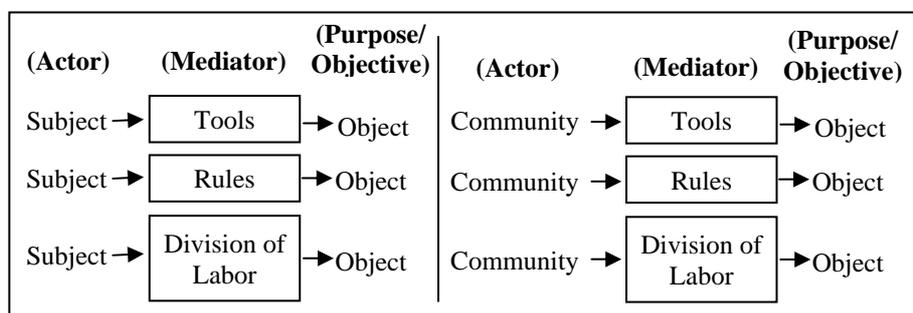
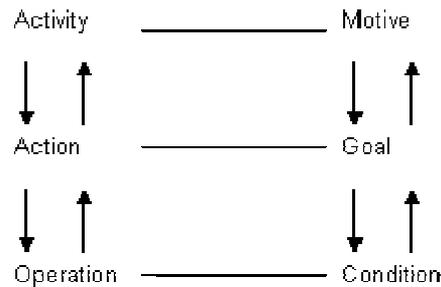


Figure 3. Adapted Activity Notations

Activities should not be considered in isolation because every activity is influenced by other activities and changes within their environment (Kutti 1996 as cited in [10]). Thus to better elicit the requirements, we need to understand the three hierarchical levels of activities introduced by [20] which are activity, action and operation [7,10].

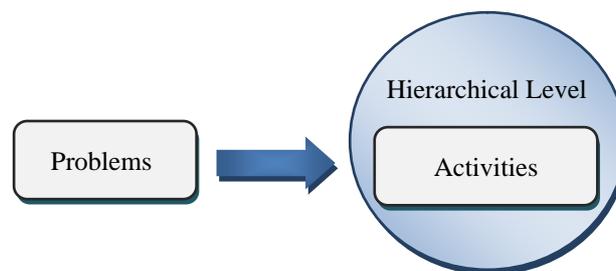


*Figure 4. The Hierarchical Levels of Activity*

As shown in Figure 4, each of this level is corresponding to motive, goal and condition accordingly. An activity can be reached through various actions, while the same action can be used as contribution to different activities. This means, every time an activity is performed or executed, a set of actions and operations will be executed too.

## 5. Towards Activity Theoretical Requirements Elicitation

Activity theory is useful in requirements gathering, especially for interactive systems [31]. The components of the activity systems such as subjects, community and objects do not directly act on each other, but they are mediated by tool(s) which is believed can streamline the development process. Here, the new requirements elicitation process based on activity theory is proposed and the process is modeled using Figure 5.



*Figure 5. The Proposed Requirements Elicitation Process Model*

The elicitation process is initiated by mapping the problems with activity theory structure to clearly address the situation; mapping the agent, tools, community, division of labor and rules accordingly. The idea should be mapped accordingly to acquire good result. Additionally, the problem is then broken down into activities, where the activities should be composed in its hierarchical levels (as shown in Figure 4). The motive for

each activity should be clearly defined as it will reflect the goal of each action. The operation is depending on the temporary goal set by the action.

To better understand the process, consider a basic example of ‘applying leave by a lecturer’ as presented below.

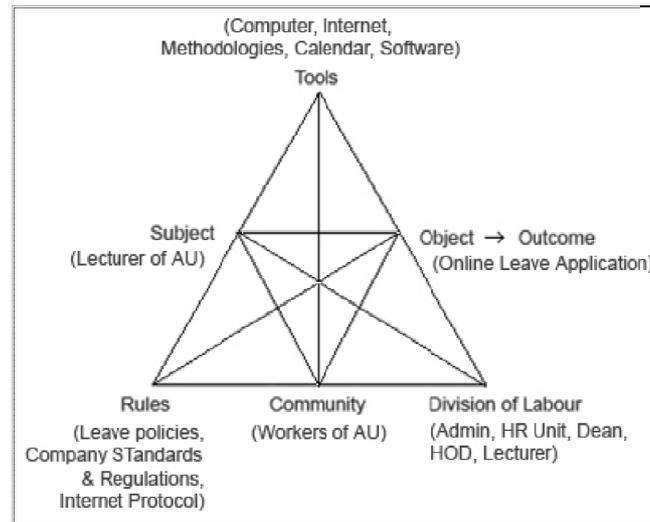


Figure 6. Analyze the Problem in the Context of Activity (An Example)

As shown in Figure 6, the problem highlighted by the Lecturer at A University (AU) is to create an online leave application for only lecturers. The problem is first analyzed in the context of activity and should be composed to its hierarchical level as shown in Table 3.

Table 3. The Hierarchical Levels for the ‘Apply Leave’ Activity (An Example)

Activity	Actions	Operations
Apply leave	Completing form	Filling in various fields in the form
		Select date
		Choose leave type
		Upload attachment(s)
	Submit form	Send form to HOD

## 6. The Impact of Activity Theory on the Requirements Elicitation Process for Web Application

In general, activity theory seems to reduce the difficulties for software requirements elicitation. However when considering the development of the Web-based system, as discussed in Section 3, few other distinctive issues need to be considered. As highlighted in Section 3.1, [6] has suggested the solution for (a) *navigational* issues; where they mentioned that mapping each element to its correlated elements can help in solving this problem because the navigational structure of the web application can be prepared before the development take place. The activity structure is also said as able to

solve the *(b) organizational* issues; by delineating the human interaction, not only with the tools and artifacts, but how they interact with the community around them as well because the social process cannot be understood properly when it analyzed separately from the external entities.

Since the solution for the following issues has not been found, hence this paper will outline some suggestions to solve those issues. The potential use of activity theory in improving the requirements elicitation process for Web application development will be outlined in the following paragraph and using Table 4 and Figure 7.

*(c) Complexity*

In their study to model the context awareness using activity theory, [32] did mention about the use of the activity theory to portray the problems in piece. Figure 3 and Table 3 serve the examples on how to make large problems become smaller (by viewing them in units). This shows the strength of the activity theory and its capability of describing each problem in piece which yield to a better understanding on specific issue. This will help on determining the best content or Medias to be placed on the Web to avoid placing the wrong Medias and content that will increase the unnecessary complexity of the Web application to make it usable and less complex.

*(d) Accessibility*

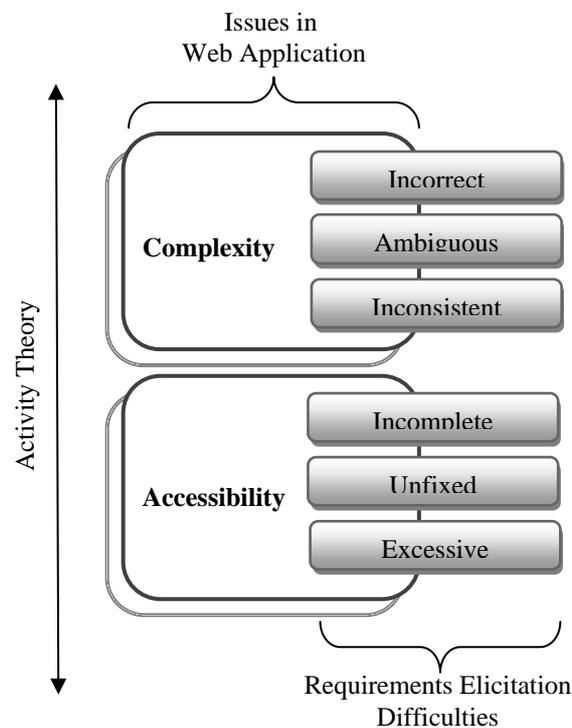
A good look of system design depends on the presentation of its requirements. Producing an easy-to-understand requirement (without ignoring the core activities) will save the time of the designers to transform the requirements into design [32]. Hence, the process should start with elicitation process in order to ensure that only required requirements are delivered to the designers. Activity theory helps the designer to understand what pieces of knowledge are involved and what are the social and technological contexts used when solving problem(s). Directly it solves the incomplete and excessive issues that make the elicitation process tougher.

With reflect to Table 1 which highlights the difficulties in requirements elicitation, the following Table 4 outline the ways of reducing such factors using activity theory and the proposed model.

**Table 4 The Potential of Activity Theory to Reduce the Requirements Elicitation Difficulties for Web Application Development**

Difficulties	How Activity Theory Helps
Incomplete	The requirements will be incomplete if the practitioner do not understand the real users' needs and do not know how to accordingly relate different user's needs to its environment. Activity theory helps to overcome the incomplete issues by addressing each problem to its own situation, which can lead to a better understanding of specific issues. This approach will help the practitioner to better understand the users' needs and the domain knowledge. This will increase the users' satisfaction and results to a better <b>accessibility</b> .
Incorrect	Activity Theory is a descriptive tool to help understand the unity of consciousness and activity. A more precise and correct requirements will be gained with the use of 'object' because the objective/purpose for each activity is well-defined in a clear, less complicated way. Thus avoiding the incorrectness. Correct and well-defined requirements will help resolving the <b>complexity</b> problems.
Ambiguous	To reduce the <b>complexity</b> issues requires the practitioner to understand each content or terms adequately. All the terms and notions can be clarified through extensive analysis within and between activities. Activity theory can give a 'vocabulary' to let the practitioner understand things easily because it considers both individual and social levels of the human practices in the development process, thus it will reduce the <b>complexity</b> of the requirements.
Inconsistent	The <b>complexity</b> of the requirements often come from the inconsistent requirements gathering, which avoid the necessary data and having conflicts with users' different view. This can be overcome by particularly focus on every 'subject' and 'community' of each problem, focus on every <b>community</b> by grouping all the <b>subjects</b> with the same <b>objects</b> , then streamlines their objective one by one using activity theory. This will ensure the important information will not go missing.
Unfixed	The <b>accessibility</b> of the requirements can be gained by avoiding the requirements change. To avoid later changes in requirements, the practitioner needs to completely gain the faultless requirements before the development starts. Using the activity theory model as proposed in Figure 5, which supports the detail problem analysis with <b>hierarchical</b> activity level will promise a fix solution. When the user satisfies about their <b>accessibility</b> , this will stop them from requesting for more changes.
Excessive	The structure of activity theory will help the practitioner keep in line while asking questions (ask only related questions within the scope, by considering the <b>subject</b> , <b>object</b> and <b>community</b> with their affected <b>mediators</b> according to its <b>activity</b> level). This concept will affirm you to get only desired information that is needed. Avoiding the excessive requirements will help improving its <b>accessibility</b> (by producing easy-to-understand requirement). And this is very important in ensuring a better system design in order to satisfy the users need.

The following Figure 7 illustrate the outcomes discussed in Table 4 which displays the relationship between the difficulties of the requirements elicitation (complexity and accessibility) and which Web application issues that they are related to.



**Figure 7. The Use of Activity Theory to Reduce the Difficulties of Requirements Elicitation in Relation to the Web Application Issues**

Since the activity theory considers both individual and social levels of the human practices in the development process, thus it can ease the requirements elicitation process. As a result, it helps the practitioners explore various issues in numerous ways by considering the involvement of external factors and environment which reduce the level of uncertainty in requirements gathering.

## 7. Conclusion

In order to get adequate and quality software requirements, it is necessary to understand the activities performed by the agents involved in the system. The motivation of this work is to overcome the essential difficulties in requirements elicitation processes, especially for Web application. Looking at the end users as the spot view in requirements elicitation process, who also has close relationship with the motives and activities, therefore, the involvement of these agent is unvoiced. Therefore, the use of activity theory that supports the human involvement is relatively useful in reducing the requirements elicitation difficulties, and at the same time is seen as potential agent to avoid the complexity and accessibility issues rise in Web application. Thus, it can be concluded that the capability of activity theory to offer better approach of eliciting the requirements will produce requirements that are better and more in quality, which in turn will lead to a better Web design.

## 8. References

- [1] Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M. & White, B. Web Engineering. *Journal of Web Engineering*, vol. 1, no. 1, pp. 003-017 (2002).
- [2] Escalona, M.J. & Koch, N. Requirements engineering for Web applications: a comparative study. *Journal on Web Engineering*, vol. 2, no. 3, pp. 193-212 (2004).
- [3] Lowe, D. Web system requirements: an overview. *International Journal on Requirements Engineering*, vol. 8, no. 2, pp. 102-113 (2003).
- [4] Bolchini, D. and Paolini, P. Capturing Web Application Requirements through Goal-Oriented Analysis, in *Proceedings of WER, 2002*, pp.16-28 (2002).
- [5] Overmyer, S.P. What's different about requirements engineering for Web sites? *International Journal on Requirements Engineering*, vol. 5, no. 1, pp. 62-65 (2000).
- [6] Uden, L., Valderas, P. and Pastor, O. (2008). An Activity Theory Based Model to Analyse Web Application Requirements. *Information Research*, vol. 13, no. 2. (2008).
- [7] Vakkayil, J.D. Activity Theory: A Useful Framework for Analysing Project-Based Organizations. *Vikalpa: The Journal for Decision Makers*. vol. 35, no. 3, pp. 1-18 (2010).
- [8] Buren, J.V. & Cook, D. Experiences in the Adoption of Requirements Engineering Technologies, CROSSTALK, *The Journal of Defense Software Engineering*, vol. 11, no. 12, pp. 3-10 (1998).
- [9] Reubenstein, H. & R. Waters. The Requirements Apprentice: Automated Assistance for Requirements Acquisition, *IEEE Transactions on Software Engineering*, vol. 17, no. 3, pp. 226-240 (1991).
- [10] Uden, L., Kumaresan, A., Salmenjoki, K. Usable Collaborative Email Requirements Using Activity Theory. *Informatica* 31, pp. 71-83 (2007).
- [11] IEEE 90 - Institute of Electrical and Electronics Engineers. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries* (1990).
- [12] Nuseibeh, B. and Easterbrook, S. Requirements Engineering: A Roadmap. *Proceedings of the Conference on The Future of Software Engineering, International Conference on Software Engineering, Limerick, Ireland*. IEEE Computer Society, pp. 35-46 (2000).
- [13] Ferreira, M.J., Loucopoulos, P. Organisation of Analysis Patterns for effective Re-use. *Proceedings of the International Conference on Enterprise Information Systems, ICEIS, Setubal, Portugal* (2001).
- [14] Davey, B. & Cope, C. Requirements Elicitation – What's Missing? *Issues in Informing Science and Information Technology, Volume 5* (2008).
- [15] Tsumaki, T. & Tamai, T. Framework for matching requirements elicitation techniques to project characteristics. *Software Process Improvement and Practice*, vol. 11, pp. 505-519 (2006).
- [16] Hickey, A.M., Davis, A.M., Kaiser, D. Requirements Elicitation Techniques: Analyzing the Gap between Technology Availability and Technology Use. *Comparative Technology Transfer and Society*, vol. 1, no. 3, pp. 279-302 (2003).
- [17] Browne, G. J. & Rogich, M. B. An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. *Journal of Management Information Systems*, vol. 17, no. 4, pp. 223 (2001).
- [18] McDonald, A. & Welland, R. Web Engineering in Practice. In *Proceedings of the 4th Workshop on Web Engineering (in conjunction with 10th International Conference on WWW), Hong Kong, May 21-30 (2001)*.
- [19] Vygotsky, L.S. *Mind in Society*. Harvard University Press, Cambridge, MA (1978).
- [20] Leont'ev, A.N. *Activity, consciousness and personality*. Englewood Cliffs, NJ: Prentice Hall (1978).
- [21] Engeström, Y. *Learning by Expanding an Activity-theoretical Approach to Development Research*. Orienta-Konsulti Oy, Helsinki, Finland (1987).
- [22] Barthelmeß, P. and Anderson, K.M. A View of Software Development Environments Based on Activity Theory. *Journal of Computer Supported Cooperative Work - CSCW* , vol. 11, no. 1-2, pp. 13-37 (2002).
- [23] Martins, L.E.G. & Daltrini, B.M. An Approach to Software Requirements Elicitation Using Precepts from Activity Theory. In *14th IEEE International Conference on Automated Software Engineering*, pp. 15-23 (1999).
- [24] Hickey, A.M. and Davis, A.M. A Unified Model of Requirements Elicitation. *Journal of Management Information Systems*, vol. 20, no. 4, pp. 65-84 (2004).

- [25] Probert, S.K. Requirements engineering, soft system methodology and workforce empowerment. *International Journal on Requirements Engineering*, vol. 4, no. 2, pp. 85-91 (1999).
- [26] Hickey, A.M. and Davis, A. M. Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes. *Proceedings: 36th Hawaii International Conference on System Sciences (HICSS'03)*, Hawaii, IEEE Computer Society (2003).
- [27] Rajagopal, P., Lee, R., Ahlswede, T., Chiang, C.C., and Karolak, D. A New Approach for Software Requirements Elicitation. *Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05)*, IEEE (2005).
- [28] Christel, M.G. & Kang, K.C. Issues in Requirements Elicitation. Technical Report. CMU/SEI-92-TR-012, ESC-TR-92-012. (1992).
- [29] Leont'ev, A.N. *Problems of the Development of Mind*. Moscow: Progress (1981).
- [30] Foot, K.A. Cultural-Historical Activity Theory as Practical Theory: Illuminating the Development of a Conflict Monitoring Network. Published in *Communication Theory*, vol. 11, no. 1, pp. 56-83 (2001).
- [31] Brown, Robert B.K; Hyland, Peter; and Piper, Ian C. Specifying System Requirements Using the 5S Method. *ACIS 2006 Proceedings*. Paper 100 (2006).
- [32] Kofod-Petersen, A. and Cassens, J. Using Activity Theory to Model Context Awareness. In Thomas R.R., Schulz, S. and David B.L. *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers, LNCS*, vol. 3946, pp. 1-17, Springer (2006).

