A Cognitive Approach to Improve Software Engineering Processes

Nadina Martinez Carod
namartin@uncoma.edu.ar

Adriana Martín
martinae@jetband.com.ar

Gabriela N. Aranda
garanda@uncoma.edu.ar

Alejandra Cechich
acechich@uncoma.edu.ar

Departamento de Ciencias de la Computación – Universidad Nacional del Comahue
Buenos Aires 1400, Neuquén, Argentina. Fax: (+54) 299-4490313

Abstract

Cognitive Informatics is a new research area that combines concepts from cognitive sciences and informatics. Particularly, classification according to cognitive styles or learning styles is a common practise in educational and business areas in order to obtain better performance from people involved in learning and collaborative tasks. Human-intensive processes in software engineering make it similar to those areas. In this paper, we propose using a kind of people classification to improve those processes of software engineering where human behaviour is a critical influence on their success.

1. Introduction

Software engineering is about "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" [16]. The goals of software engineering activities are maximizing productivity, minimizing cost, and controlling the quality of both, processes and resultant products [18].

Most processes in software engineering have the characteristic of requiring an intensive communication between stakeholders with different backgrounds, believes, culture, etc., so that applying well-known techniques from the field of cognitive sciences may help in finding good strategies to deal with such a diversity.

Cognitive Informatics is a profound interdisciplinary research area that combines cognitive science and informatics. One of the most interesting things found in cognitive informatics is that it embodies many science and engineering disciplines, such as informatics, computing, software engineering, and cognitive sciences, which share a common root problem: *how natural intelligence processes information* [4, 17].

Cognitive informatics definition says that it is "an extension of contemporary informatics into the study of the brain and its information processing mechanisms" [4]. It is related to activities of information processing in the brain such as information acquisition, representation, memory, retrieval, generation, and communication [17].

Many research areas in computer sciences contribute to cognitive sciences, like artificial intelligence, knowledge management, neural networks, etc. In software engineering there are also processes that can be related to cognitive informatics, especially those that concern software complexity and program comprehension processes [4].

There are some basic characteristics that determine the complexity of software development from the point of view of cognitive informatics that have been presented in [18]. Some of them are, for instance, the difficulty of establishing and stabilizing requirements; the necessity of varying domain knowledge; the dependability of interactions between software, hardware and human beings.

Cognitive psychology studies the thinking mind and the mental processes concerning the way people attend and gain information and how these information processing mechanisms affect human behaviour [4]. Cognitive styles and learning styles models are based on such observations. Particularly, cognitive styles are based on Jung's theory published in 1921, which classifies people preferences about perception, judgment and processing of information [13]. This classification is used to analyse and understand differences in human behaviour. Different instruments have been designed to measure human characteristics and explain their differences.

Similarly, learning style models (LSM) classify people according to a set of behavioural characteristics pertaining to the ways they receive and process information. They are used to improve the way people learn a given task. For example, the learning style model proposed by Felder-Silverman [7] classifies people into four categories, each of them further decomposed into two subcategories: Sensing-Intuitive; Visual-Verbal; Active-Reflective; Sequential-Global. People are classified by a multiple-choice test, available in the WWW, which gives them a rank for each subcategory.

Even when LSM have been discussed in the context of analysing relationships between instructors and students, we can take advantage of this kind of models and adapt it to virtual teams that deal with some software engineering processes. To do so, we consider an analogy between stakeholders and roles in LSM, since during most of the processes in software engineering everybody "learns" from others. In this way, stakeholders play the role of student or instructor alternatively, depending on the moment or the task they are carrying out [12].

With this in mind, section 2 introduces three processes we are working on, and sets aspects that make "cognitive application" a possible solution to human communication problems. Conclusions and future work are addressed in the final section.

2. Cognitive Approaches to Improve Software Engineering

In our research, we apply techniques, like the learning styles classification presented before, to improve some processes that belong to the field of Software Engineering.

The processes we are currently working on are:

- Elicitation process in a distributed environment;
- Requirement negotiation process between stakeholders involved in a software project;
- Re-engineering of business processes.

The characteristic that these processes share is the important role that human communication plays in all of them. Main characteristics of these processes, and the current state of our research in each one, are explained below.

2.1 Distributed Elicitation Process

It is a common practise these days that software development processes are carried out in distributed environments, meaning that stakeholders are in many geographically distanced sites. This kind of multi-site or global software development is currently matter of study and discussion.

Problems during traditional requirement elicitation processes have been detected and analysed for [6, 10]. Moreover, when participants are distributed, distance affects processes of communication, coordination and control, and has consequences along all the software development process [5], specially during requirements elicitation, which is critically based on communication between stakeholders [16].

The CSCW (Computer-Supported Cooperative Work) is a research area that focuses on experimental systems and workspaces [8] and studies human behaviour as well as the technical support the group needs to work in a more productive way. Groupware is the software designed to support the group interaction and facilitate the communication between the participants.

In this area, our work focuses on the application of Learning Style Models (LSM) to enhance interpersonal communication in geographically distributed teams during the elicitation phase. In order to do so, we have proposed an intuitive classification of groupware tools according to the main characteristics of each category on the Felder and Silverman model [12]. Later we have proposed a model based on fuzzy logic and fuzzy sets to obtain, from a set of examples, a set of rules that tell us about people preferences for groupware tools according to their classification in the Felder Silverman Model [1]. Following, in [2], we have presented a model and a prototype tool to automate the selection process of groupware tools and elicitation techniques. Future work in this area is defining a methodology for requirement elicitation in distributed environments focusing on the personal characteristics of stakeholders.

2.2 Requirement Negotiation

Usually, every software development deals with incompatibilities between stakeholders' priorities. In case of diverging opinions between stakeholders exist, they must work out to develop acceptable solutions for all people involved. This fact implies a negotiation process to balance conflicting requirements. From this point of view, negotiation may be seen as the key of a successful software projects development. Besides, often developers cannot implement all requirements because of time and resource constraints. Instead, they focus on implementing firstly the requirements that are set as the most important.

The process of negotiation involves both prioritizing requirements, and selecting the set of requirements to be satisfied. So far, relatively few research efforts have been done to get mechanisms for prioritizing requirements. Some examples are the Quality Attribute Requirements and Conflict Consultant tool [3] within the Barry Boehm's WinWin System [14], and the Cost-Value method based on analytical techniques [9] by J. Karlsson and Ryan, which use the Analytic Hierarchy Process (AHP) [15]. Two disadvantages are detected when applying those methods: (1) the negotiation process becomes difficult when the number of stakeholders increases; and (2) only one stakeholder has the responsibility of estimating the relative requirements value.

In order to improve the prioritization process, we propose measuring and weighting each stakeholder' cognitive attributes. The assignment of cognitive weights to each stakeholder may help assess a candidate group to be involved in prioritizing a set of requirements. Therefore, the requirements prioritization process can be adjusted based on stakeholders' profiles using cognitive aspects of stakeholders.

Our objective is defining templates for developing a robust tool-supported system, to be used in any negotiation process.

2.3 Re-engineering of business processes

Changing a business process imply intensively gather and manipulate knowledge from both machines and human beings. In this context, the success of a Business-Process Reengineering (BPR) certainly requires evolved techniques to support stakeholders' cognition. Such techniques may help to identify and manage all existing relevant knowledge efficiently.

The scene described in the previous paragraph gave us a basis and a motivation for studying and investigating cognitive theories and models and their relation with BPR approaches. Based on the statement that human cognition has been always a decisive attribute when re-engineering work has to be done, we believe that substantial improvements can be achieved discovering ways of applying Cognitive Informatics concepts to share and understand knowledge in re-engineering a knowledge-intensive business process.

The main challenge implies to determine how to apply cognitive aspects for effective and dynamic knowledge-based practices. On this line, we have recently identified some cognitive aspects used by traditional and new reengineering models and we have provided a framework highlighting how cognitive aspects might improve reengineering through knowledge and perception modelling [11].

The next step is exploring how to transform cognitive issues into a dynamical knowledge instrument for process improvement, looking for a Cognitive Reengineering discipline that facilitates people cognition involved in the process.

3. Conclusions and Future Work

Software engineering processes imply dealing with complex interactions between stakeholders. Consequently, software engineering is a cognitive work domain were human cognition plays a critical role. Cognitive sciences study how personal preferences affect communication and learning processes. These kinds of considerations have been applied for decades in educational and business fields to improve the way people interact with each other and to take advantage of their personal characteristics to increase their performance in individual and group work.

In our research, we propose analysing in which way those concepts and techniques can be applied to increase the quality of some software engineering processes, where communication is crucial. Doing so, we have a particular interest in giving answers to questions related to how stakeholders may find a common understanding of the final product they try to define.

Current work is done in analysing cognitive aspects of such processes in order to propose new methodologies based on cognitive techniques. Future work is needed in experimenting and validating such methodologies.

4. References

[1] Aranda, G., Cechich, A., Vizcaíno, A., and Castro-Schez, J.J. Using fuzzy sets to analyse personal preferences on groupware tools. In *X Congreso Argentino de Ciencias de la Computación, CACIC 2004*, San Justo, Argentina, p. 549-560, October 2004.

[2] Aranda, G., Vizcaíno, A., Cechich, A., and Piattini, M. A Cognitive Perspective for Choosing Groupware Tools and Elicitation Techniques in Virtual Teams. In *International Conference on Computational Science and its Applications (ICCSA 2005)*, Singapore, May 2005.

[3] Boehm, B. and In, H., Identifying Quality -Requirements Conflicts. IEEE Software, Vol. 13(2): p. 25-35, March 1996.

[4] Chiew, V. and Wang, Y. From Cognitive Psychology to Cognitive Informatics. In *Second IEEE International Conference on Cognitive Informatics, ICCI'03*, London, UK, p. 114-120, August 2003.

[5] Damian, D., Lanubile, F., Hargreaves, E., and Chisan, J. Workshop Introduction. In *3rd International Workshop on Global Software Development. Co-located with ICSE 2004*, Edinburgh, Scotland, May 2004.

[6] Davis, A., Software Requirements: Objects, Functions and States. Upper Saddle River. New Jersey: Prentice Hall, 1993.

[7] Felder, R. and Silverman, L., Learning and Teaching Styles in Engineering Education. Engineering Education, Vol. 78(7): p. 674-681, April 1988.

[8] Grudin, J., Computer-Supported Cooperative Work: History and Focus. IEEE Computer, Vol. 27(5): p. 19-26, May 1994.

[9] Karlsson, J. and Ryan, K., A Cost-Value Approach for Prioritizing Requirements. IEEE Software, Vol. 14(5): p. 67-74, September/October 1997.

[10] Loucopoulos, P. and Karakostas, V., System Requirements Engineering. International series in Software Engineering, ed. Mc Graw-Hill. New York, NY, USA, 1995.

[11] Martín, A. and Cechich, A. "Identifying Cognitive Aspects to Improve Business Process Reengineering". In *X Congreso Argentino en Ciencias de la Computación, CACIC 2004*, San Justo, Argentina, p. 284-295, October 2004.

[12] Martín, A., Martinez, C., Martinez, N., Aranda, G., and Cechich, A. Classifying Groupware Tools to Improve Communication in Geographically Distributed Elicitation. In *IX Congreso Argentino de Ciencias de la Computación, CACIC 2003*, La Plata, Argentina, p. 942-953, October 2003.

[13] Miller, J. and Yin, Z., A Cognitive-Based Mechanism for Constructing Software Inspection Teams. IEEE Transactions on Software Engineering, Vol. 30(11): p. 811-825, November 2004.

[14] Park, J., Port, D., Boehm, B., and In, H. Supporting Distributed Collaborative Prioritization for Win Win requirements Captures and Negotiations. In *World Multiconference on Systemics, Cybernetics and Informatics*, Vol. 2, p. 578-584, July 1999.

[15] Saaty, T.L., The Analytic Hierarchy Process. New York: McGraw-Hill, 1980.

[16] SWEBOK, Guide to the Software Engineering Body of Knowledge, ed. Software Engineering Coordinating Committee (IEEE-CS y ACM), 2004.

[17] Wang, Y. On Cognitive Informatics. In *First IEEE International Conference on Cognitive Informatics, ICCI'02*, Calgary, Alberta, Canada, p. 34-42, August 2002.

[18] Wang, Y. On the Cognitive Informatics Foundations of Software Engineering. In *Third IEEE International Conference on Cognitive Informatics, ICCI'04*, Victoria, Canada, p. 22-31, August 2004.