

Chapter 94

The Use of HCI Approaches into Distributed CSCL Activities Applied to Software Engineering Courses

Fáber D. Giraldo

University of Quindío, Colombia

María Lili Villegas

University of Quindío, Colombia

César A. Collazos

University of Cauca, Colombia

ABSTRACT

This chapter is written as one method to supply the necessary support systems for educational and training design. As such, the authors propose their global development software (GDS) methodology emerges as a revolutionary discipline. It is based on the externalization of software development between geographically distant places in order to reduce development costs. Traditional educational and training process in software engineering must be advocated to consider (or enhance) this new trend, with its respective challenges and necessary skills (multicultural interaction, effective communication, distributed software project management), into curriculums. GDS therefore demands the presence of supporting systems to provide permanent user interaction and enhanced communication tasks. The presence of such interactions is a key aspect to promote the performance and knowledge acquisition processes among globally distributed software development teams. The main goal of such interactions into platforms that support distributed contexts is to reduce the impact generated by the tyranny of distance. This work exposes some human-computer interaction (HCI) principles applied by the authors' research team in order to structure a supporting user interface environment that reflects the distributed computer supported collaborative learning (CSCL) practice in software engineering. The chapter describes several services that are provided for managing the interaction between participants, such as synchronous interactions through Microsoft © LiveMeeting and Adobe © Connect, and asynchronous interactions such as Moodle forums. In this way, the authors implement effective HCI into educational professional practice scenarios for a distributed CSCL within the specialized domain of software engineering.

DOI: 10.4018/978-1-4666-4301-7.ch094

INTRODUCTION

Since 2008, we have created a consortium of Latin American universities with the purpose of applying distributed computer supported collaborative learning (CSCL) experiences in software engineering courses supported by high speed academics networks of Latin America. This consortium is called Latin American colaboratory of eXperimental software engineering research (LACXSER) (<http://www.lacxser.org/>). The consortium of Latin American universities involved in this project has proposed a collaborative model to support the distributed teaching learning software engineering process called: collaborative distributed learning activity (CODILA) (Collazos et al., 2010).

This experimented instructional environment proposes to include learning activities through collaborative work. To this end, the structure and dynamics of the model try to involve participants, who are geographically dispersed, so that the development of their activities can be achieved in an effective manner. Also, the model promotes the active participation of members of the workgroups. This technology provides academies with high speed networks of each Latin American country (such as the videoconference from RENATA <http://www.renata.edu.co/>- offering low cost, custom software solutions for various educational organizations, in Colombia for example), and developed by research from open source initiatives such as Moodle and Jommla.

In this chapter we will provide a background to set the context for our approach for the use of HCI as the means to implement distributed CSCL activities as they apply to 'software engineering' courses. These distributed interactions are then further explained showing how CSCL works well. Central to our HCI approach is the importance of including developmental strategies that involve usability accessibility; we provide a more detailed description of how we conducted our study to evaluate our newly developed tool. This description includes: the information archi-

ture; the interaction design; the usability tests; and the technological platforms that were used to support the CSCL.

BACKGROUND

Collaborative learning is a pedagogical framework that has been interested in the study and reflection of the real dynamic of learning (Johnson & Johnson, 1986). It can inquire about the educational interactions because it is supported by cognitive theories: situations in which the protagonists (or users) are acting and interacting simultaneously in particular contexts about a specific task or the learning content, in order to achieve more or less defined goals. The teaching/learning process is determined by communication and the features of the educational context and interpersonal contact in the development of teaching. We propose that the CSCL is characterized by the equality that each individual must have in the process, and mutuality (connection, depth and scope bi-directionality that reach the experience), this variable is a function of an existing competitive level, which involves the distribution of responsibilities, joint planning and exchange of roles (Stahl, 2010).

This educational interaction is a prominent feature of collaborative distributed teaching in software engineering, as it seeks out how to provide spaces in which it is possible to develop individual and group skills from the discussions and interactions among the students when they explore new concepts. Each student is responsible for their own learning, generating discussions among themselves when they explore concepts for elucidating, or troublesome situations to be solved. Therefore, looking for the combination of situations and social interactions may contribute to personal and significant group learning.

In a modern software engineering teaching and learning process, students cannot learn alone: the auto-structuring activity of the subject is mediated by the influence of others. Therefore, learning is a

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/use-hci-approaches-into-distributed/77789

Related Content

How Can We Trust Agents in Multi-Agent Environments? Techniques and Challenges

Kostas Kolomvatsos and Stathes Hadjiefthymiades (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 2843-2864).

www.irma-international.org/chapter/can-trust-agents-multi-agent/29539

Depression Detection Model in Social Network Content: A Survey

Esraa M. Rabie, Atef F. Hashem and Fahad Kamal Alsheref (2022). *International Journal of Software Innovation* (pp. 1-12).

www.irma-international.org/article/depression-detection-model-in-social-network-content/309112

Intelligent Fault Diagnosis for Bridge via Modal Analysis

Wenjun Zhuang (2022). *International Journal of Information System Modeling and Design* (pp. 1-12).

www.irma-international.org/article/intelligent-fault-diagnosis-for-bridge-via-modal-analysis/313582

Expressing and Validating OCL Constraints using Graphs

Najet Zoubeir, Adel Khalfallah and Samir Ben Ahmed (2013). *Embedded Computing Systems: Applications, Optimization, and Advanced Design* (pp. 93-107).

www.irma-international.org/chapter/expressing-validating-ocl-constraints-using/76952

The Impact of Social Media in Web-Based Information Management and Digital Media

Monika Steinberg (2013). *Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences* (pp. 432-453).

www.irma-international.org/chapter/impact-social-media-web-based/70621