Designing a Simulator for the Training of Software Engineers in Global Requirements Elicitation

Miguel Romero
University of Bío-Bío,
Department of Computer Science and Information Technologies, Avenida Andrés Bello s/n 3780000 Chillán, Chile.
mromero@pehuen.chillan.ubiobio.cl

Aurora Vizcaíno, Mario Piattini
University of Castilla-La Mancha
Alarcos Research Group- Institute of Information Technologies & Systems
Dep. of Information Technologies & Systems - Escuela Superior de Informática
{Aurora.Vizcaino, Mario.Piatini}@uclm.es

Abstract

The requirements elicitation process is particularly difficult in Global Software Development (GSD) environments owing principally to cultural differences and communication problems derived from the geographical distance that separates stakeholders. For this reason it is necessary to train professionals in the skills needed to confront a requirements elicitation process in a GSD environment. In this paper we present the generic and specific competences derived from our review of literature that a software engineer must have if s/he is to carry out requirements elicitation. We have used these skills to design a tool to support the teaching of requirements elicitation in GSD. This tool is a simulator which, by using virtual agents, will enable students and professionals to acquire a subset of the skills necessary for requirements elicitation in GSD.

1. Introduction

Global Software Development (GSD) [1, 2] is one of the current trends in software engineering. GSD has grown considerably in recent years due to the culture of globalization [3, 4] and other factors such as off-shoring [5, 6], and it will continue to do so, as for businesses it implies a means through which to reduce development costs whilst maintaining the level of quality [7]. However, its practice is causing some problems, which are widely referred to in literature [8] (i.e. cultural difference, inadequate communication, time difference and trust).

With the arrival of GSD, the teaching of requirements elicitation must be adapted to fit to changes in industry.

One challenge in the teaching of requirements elicitation in GSD is that of succeeding in giving the student the chance to learn from concrete experiences which are closer to real work. To do this, it is necessary to develop teaching strategies for active and collaborative learning [9], with which students learn by doing instead of just listening to an expert talking about his/her experience.

The techniques that are commonly used with such an aim are: project-based learning [10], in which students participate in teams, in order to solve a problem; and role playing games [11-13], in which students play a role (software engineer, client, user, analyst, etc.) in a simulated requirement elicitation scenario.

Different strategies have been defined to confront the challenge of teaching and training software engineers to work on GSD projects. These are: curricular changes [14-18], a closer interaction between industry and the academic world [19], joint software projects between universities from different countries [20, 21], and postgraduate specialization [22].

However, there are certain problems in putting these strategies into practice such as the difficulty in finding companies who are willing to invest time and resources in a joint education project with universities, or the students' lack of experience, which may be a very high risk factor for real projects.

In order to obtain professionals who are trained in the process, and who are capable of accomplishing top-quality requirement elicitation and even confronting the difficulties of GSD environments, the

various dimensions of teaching must be adjusted: contents, learning tools, learning techniques, assessment strategies, learning outcomes and professional competences.

In this context, defining the professional competences which are necessary for software engineers to be able to work in GSD environments is fundamental, since it will allow us to have clear ideas about the professionals the industry needs so as to define the correct contents, learning tools and the other dimensions we have previously mentioned.

In this work we present the generic and specific competences derived from our review of literature that a software engineer must have if s/he is to carry out requirements elicitation. We have used these competencies to design a simulation environment that permits the support of the training of engineers in the global requirements elicitation process.

This simulator may be a prior step towards students' participation in real projects developed between universities and the GSD industry. In fact, the experience acquired with the simulator would diminish the risk of non-qualified people being involved in real projects. Furthermore, a virtual industrial partner is provided by means of the simulator for universities that do not have one.

In the following section we describe the virtual environment that we propose, and the competencies that will be supported. Section 3, shows the technology for implementation, Section 4 describes related works, and in Section 5 we present our conclusions.

2. Proposed simulator

We propose a simulator of the requirements elicitation process in the global context in which the student (taking on the role of a requirements engineer) interacts with various stakeholders which will be virtual humans and/or real humans, in order for them to obtain the functional and non-functional requirements of the software to be developed.

Simulation is a technique which has been used in teaching for many years. It is successfully used, for example, in medicine [23] and aviation. Its main advantage is that it allows students to train themselves without the risk of a real environment, and at a lower cost.

2.1 Management scenarios

The simulator will allow the professor to create new scenarios, indicating the description of the scene, the virtual agents to be used, his personality and culture.

The lessons are aimed at specific issues of requirements elicitation in GSD in order to confront students with difficult common situations.

The scenarios can be grouped into modules in order to facilitate their administration. The teacher can create, modify and delete training scenarios and educational modules.

2.2 History Review

Initially, the students must enter their data with the aim of tracking the learning process. The system must then show the different lessons or units that it has developed, showing their results. In addition, the system must permit a review of the history of talks with each of the stakeholders.

The teacher must access this information for each student, along with a statistical summary of the evaluations obtained by each of his/her students in a given course.

2.3 Execution of training scenario

Another capability of the system will be to show the lessons that it has not developed, allowing the student to select any of them.

Figure 3 presents the steps for the development of a scenario.

When students perform a lesson selection, the system must submit the context of the problem in which the elicitation is developed and show the participant stakeholders and their roles.

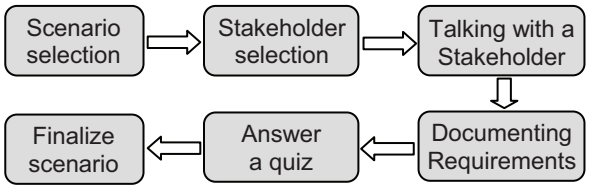

Figure 3. Steps for the development of a scenario

The students may carry out the requirements elicitation both individually or as members of a requirements elicitation team. This team may be made up either of virtual agents or humans.

Through interviews with the various stakeholders (which may be of different nationalities), the student should elicit customer requirements. We wish to enhance capacity building for developing interviews since, according to a systematic review in [24] and the empirical study in [25], this technique is more effective.

With the end of being evaluated (see Section 3.4), the student will document the requirements, respond to a questionnaire relating to the scenario which has taken place, and will then finalize the scenario.

2.4 Evaluation strategies

The students should prepare a list of requirements, both functional and non-functional, which should be sent to the system for its validation at the end of the simulation with the purpose of measuring the quality of the work done by the student. The system should provide an interface for keeping a list of the student's requirements (see Figure 4). The imput of the requirements will be limited by a structured set of sentences similar to a requirements pattern with the intention of easing automatic validation.

The requirements document will be checked to detect faults such as: ambiguous requirements, non-existent requirements, unspecified requirements, etc.

Besides this evaluation, the system will also record the questions that the student has formulated in an inadequate way with regard to the cultural differences and protocol of communication (manner of greeting and taking one's leave, degree of formality informality etc.).

In addition, the simulator will validate the student's work by means of a questionnaire which will be designed by a profesor to evaluate the knowledge that the students have acquired whilst taking part in the scenario. This evaluation is optional.

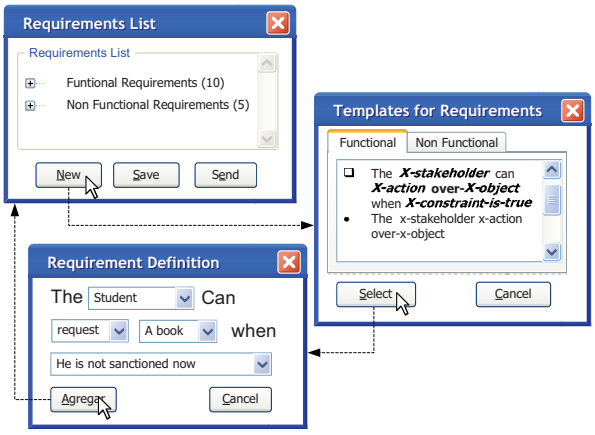

Figure 4. Prototype of requirements documentation interface

2.5 Architecture of simulator

The architecture of the simulator is divided into three layers: User interface, Services and Data access object. The aim of this separation is to facilitate the implementation of different types of user interfaces, such as desktop, or web interfaces between systems. In addition, the layer of data access objects permits the minimization of the existing coupling between the application and the DBMS used.

Figure 2 show the architecture of our simulator.

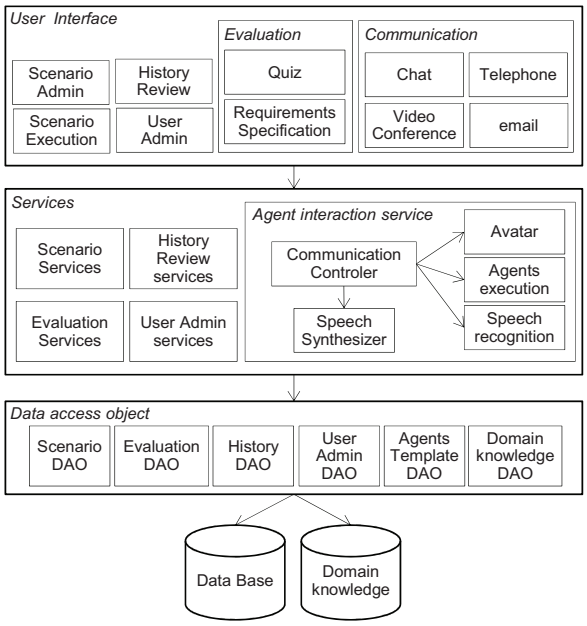

Figure 2. Architecture of the simulator.

The user interface layer is made up of the following components:

- *Scenario administrator,* which permits the creation, modification and deletion of a scene.
- *Scenario execution,* which enables scenario training.
- *History review*, which allows a review of the scenarios that have been implemented and all the information generated during its execution.
- *User administrator*, which enables the creation, elimination and modification of user accounts.
- *Evaluation*, this component allows students to complete the evaluation questionnaires, and they may enter the requirements document they have produced.
- The components of *communication*, instant messaging, Telephone, Video Conference, and email which will permit a natural communication between the student and virtual agents.

The service layer is designed to process the data obtained from the user interface layer. This layer is composed of:

- *Scenario Service*, which provides services to components: execution scenario, scenario administrator.
- *Evaluation Services*, which services the evaluation component of the user interface layer.
- *History review services*, this component provides services to the History review component of the user interface.
- *Agent interaction service*, which is the main component of our architecture. This is where input from the user during conversations with a virtual stakeholder (virtual agents) is interpreted. This component is made up of the following:
  - *Agents Execution*, this component provides answers from a particular agent to questions from the user. The input is a text, and the output is another text.
  - *Communication controller*, this component is the link between the communication interface between the user and the virtual agent. The following component parts are used according to the type of user interface with which they do or do not interact: avatar, speech recognition, speech synthesizer.
  - *Avatar,* this component generates a graphical representation of the virtual agent for use in a video conference.
  - *Speech Recognition*, this component converts audio to text by means of voice recognition. This component is used in a phone call or in a video conference.
  - Speech Synthesizer, this component converts the text into an audio file. This component is used in a phone call or in a video conference.

The persistance layer is in charge of providing the the services which are necessary to create, modify, eliminate or consult the information stored in the data base and in the knowledge base.

3.6 Competencies to be developed by the simulator

The intention of this simulator is not that of developing all the competencies which are necessary for requirement elicitation in GSD, but only a subset of them. These are:

2.6.1 The ability to prepare an interview, taking into account such factors as the differences in the stakeholders' cultural and language. The theoretical aspects of how to prepare and carry out an interview will be studied by the students befote they use the simulator. Their competency will be evaluated by reviewing the record of the conversations that have taken place between the student and the various stakeholders. Automatic evaluation will be limited to a set of patterns which this virtual agents are able to detect, and the evaluation must eventually be completed by the professor.

2.6.2 Elicitation (in a GSD environment) of real requirements based on stakeholder's needs using an Interview Technique and computer mediated Communications. The student will be trained in this competency through his/her interaction with the virtual agents. Each scenario will be directed towards a specific problem in GSD requirements elicitation, simulating the manner in which this competence is acquired in professional practice. In order to evaluate this competence, when defining the scenario the professor will indicate which are both the functional and non-functional requirements that the student should attain. The simulator will contrast the student's requirements specification with that defined by the profesor. As has previously been mentioned, the requirements definition will take place with the use of a structured language which will ease said comparison.

2.6.3 Specification of functional and non-functional requirements for different types of Systems, taking into account the quality requirements attributes. In order to attain this competence, the student must know the quality standards for requirements definition. Although the requirements specification will be structured, we are contemplating the incorporation of patterns in the simulator which will low quality requirements. The simulator will know which patterns permit it to give an automatic response concerning the quality of the request specification, although the set of patterns recognized by the simulator will be limited.

5. Related Works

Reference [26] shows an Educational Virtual Environment EVE which uses a Virtual Human with the goal of training students in the Arabic language and in Arabic cultural familiarization.

As in this environment, our proposal aims to teach these cultural differences with regard to the major cultures involved in the GSD (The West, India, and China).

Another EVE appears in [23] in which the virtual Human is used to simulate the patients who are interviewed by medical students. The interaction between the virtual patient and the students is that of talking in a natural manner.

Within the scope of education in software engineering, reference [27] presents a simulator which allows students to assume the role of a software project manager. In this simulator, the student uses a textual user interface to hire or lay off employees, and s/he may be asked to perform any tasks that are useful in software development such as preparing requirements specification, reviewing the design document, or testing the code [27]. Most of the messages obtained are statements from his/her 'employees', such as "I have completed the specification", or "During the tests I found x errors" [27]. The student must carefully review such statements and react in an appropriate manner, because this is all the information that the simulator has delivered. When the game has finished, the player receives his scorecard and domestic variables can be analyzed to evaluate his/her performance [27]. This work is similar to that which we propose but in a different context, which changes the objectives of the learning: we focus upon education in the requirements elicitation process and not upon the software project administration process.

6. Conclusions

GSD is a current trend, which greatly influences the way in which software is developed. This requires universities and the software industry to rethink the way in which software engineers are taught and trained.

In this paper we present the generic and specific competencies derived from our review of literature that a software engineer must have if s/he is to carry out requirements elicitation.

From these competencies we have designed an architecture for a tool with which to support the teaching and training of requirements elicitation in GSD. This tool is a simulator which, by using virtual agents, will enable students and professionals to acquire a subset of the skills necessary for requirements elicitation in GSD, such as: the elicitation of real requirements based on a stakeholder's needs using an Interview Technique and computer mediated communications, the ability to work in an international context, the understanding of the cultures and customs of other countries

7. Acknowledgements

This work is partially supported by the MELISA project (PAC08-0142-3315), Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain; ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/Fondos Europeos de Desarrollo Regional (FEDER) in Spain; the CompetiSoft project (506AC0287, CYTED program).

8. References

[1] J. D. Herbsleb, "Global Software Engineering: The Future of Socio-technical Coordination," presented at Future of Software Engineering(FOSE'07) at ICSE'07, Minneapolis, 2007.

[2] J. D. Herbsleb and D. Moitra, "Guest Editors' Introduction: Global Software Development," *IEEE Software*, vol. 18, pp. 16-20, 2001.

[3] B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," presented at Future of Software Engineering, 2007 (FOSE '07), 2007.

[4] B. Boehm, "The Future of Software and Systems Engineering Processes," University of Southern California, Los Angeles, CA 90089-0781, Tech Report USC-CSE-TR-507, June 2005.

[5] J. M. Bhat, M. Gupta, and S. N. Murthy, "Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing," *IEEE Software*, vol. 23, pp. 38-44, 2006.

[6] E. Carmel and P. Abbott, "Configurations of global software development: offshore versus nearshore," presented at 2006 international workshop on Global software development for the practitioner, during ICSE'06, Shanghai, China, 2006.

[7] J. Audy, R. Evaristo, and M. B. Watson-Manheim, "Distributed Analysis The Last Frontier?," presented at 37th

Annual Hawaii International Conference on Systems Sciences (HICSS), Big Island, Hawaii, 2004.

[8] D. E. Damian and D. Zowghi, "The Impact of Stakeholders' Geographical Distribution on Managing Requirements in a Multi-Site Organization," presented at IEEE Joint International Conference on Requirements Engineering (RE'02), Los Alamitos, CA, USA, 2002.

[9] D. Rosca, "An Active/ Collaborative Aproach in Teaching Requirements Engineering," presented at 30th Annual Frontiers in Education - Vol 1 (FIE'2000), 2000.

[10] R. B. J. Vaughn and J. Carver, "Position Paper: The Importance of Experience with Industry in Software Engineering Education," presented at 19th Conference on Software Engineering Education and training Workshops (CSEETW' 06), 2006.

[11] L. Jaccheri and G. Sindre, "Software Engineering Students meet Interdisciplinary Project work and Art," presented at 11 th International Conference Information visualization (IV'07), 2007.

[12] G. Sindre, "Teaching Oral Communication Techniques in RE by Student-Student Role Play: Initial Experiences," presented at 18th Conference on Software Engineering Education & Training (cseet'05), 2005.

[13] M. L. Barrett, "Simulating requirements gathering," *SIGCSE Bulletin*, vol. 29, pp. 310-314, 1997.

[14] J. H. Cross-II, "21st-Century Global Software Development and Education Position Statement," presented at 29th Annual International Computer Software and Applications Conference (COMPSAC'05), 2005.

[15] O. Minor and J. Armarego, "Requirements Engineering: A Close Look At Industry Needs And Model Curricula," *Australian Journal of Information Systems (AJIS)*, vol. 13, 2005.

[16] U. Bellur, "An Academic Perspective on Globalization in the Software Industry," presented at 30th Annual International Computer Software and Applications Conference (COMPSAC'06), 2006.

[17] V. Vasudevan, "Global Software Enterpreneurship," presented at 30th Annual International Computer Software and Applications Conference (COMPSAC'06), 2006.

[18] R. Ramnath, "Global Software Development for the Enterprise," presented at 30th Annual International Computer Software and Applications Conference (COMPSAC'06), 2006.

[19] D. Lee, K. Jung, K. Yi, Y. Cho, Y. Han, and D. Kang, "Development of partnership between industry and university via customized field-oriented curriculum," presented at 2005 IEEE International Conference on Microelectronic Systems Education (MSE'05), 2005.

[20] I. Richardson, A. Milewski, N. Mullick, and P. Keil, "Distributed development: an education perspective on the global studio project," presented at International Conference on Software Engineering (ICSE'06), 2006.

[21] S. I. Ahamed, "Model for Global Software Engineering Project Life Cycle and How to Use it in Classroom for Preparing Our Students for the Globalization," presented at 30th Annual International Computer Software and Applications Conference (COMPSAC'06), 2006.

[22] P. Lago, H. Muccini, L. Beus-Dukic, I. Crnkovic, S. Punnekkat, and H. V. Vliet, "Towards a European Master Programme on Global Software Engineering," presented at 20th Conference on Software Engineering Education & Training (CSEET'07), 2007.

[23] B. Lok, "Teaching Communication Skills with Virtual Humans," in *IEEE Computer Graphics and Applications*, vol. 26, 2006, pp. 10-13.

[24] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. Moreno, "Effectiveness of Requirements Elicitation Techniques: Empirical results Derived from a Systematic Review," presented at 14th IEEE International Requirements Engineering Conference (RE'06), 2006.

[25] W. Lloyd, M. B. Rosson, and J. Arthur, "Effectiveness of Elicitation Techniques in Distributed Requirements Engineering," presented at 10th Anniversary IEEE Joint International Conference on Requirements Engineering, RE'02, Essen, Germany, 2002.

[26] E. M. Sims, "Reusable, lifelike virtual humans for mentoring and role-playing," *Computers & Education*, vol. 2007, pp. 75-92, 2007.

[27] A. Drappa and J. Ludewig, "Simulation in software engineering training," presented at 22nd international conference on Software engineering, Limerick, Ireland, 2000.