A FRAMEWORK FOR DEFINING SIMULATORS WITH WHICH TO TRAIN GLOBAL SOFTWARE DEVELOPMENT

Miguel J. Monsor, Aurora Vizcaíno and Mario Piattini
Alarcos Research Group, Institute of Information Technologies & Systems, Escuela Superior de Informática
University of Castilla-La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, Spain

Keywords: Global Software Development, Distributed Software Development, Engineering Education, Educational Environment, Teaching Model, Simulators, Virtual Agents.

Abstract: The teaching and training of Global Software Development (GSD) entails several well-known difficulties, of which the problem of establishing environments in which students can learn by practicing in realistic scenarios is commonly reported. In this paper we propose an educational framework that uses simulation to train future participants to confront the principal problems encountered in GSD (cultural, language and communication problems). Our framework therefore provides support for the design of these simulations by means of a tool that trains its users in the typical problems that may occur during interactions between distributed members. The simulations place learners in realistic GSD scenarios in which they will interact with virtual participants, thus permitting them to confront the collaborative, organizational and technical problems of GSD.

1 INTRODUCTION

Global Software Development (GSD) is currently being addressed in educational environments with the aim of training software engineering students in the challenges that it entails. The principal problems of GSD are specifically related to the establishment of an effective communication between the distributed participants (Monasor et al., 2009) who must interact with people from different cultures and different time zones in a common language in order to jointly develop a software project.

Participants in GSD activities therefore require additional skills in order to minimize the impact of the inexperience and fears that they frequently have to confront when dealing with problematic GSD situations (Casey and Richardson, 2008), and to avoid costly delays in time-to-market.

Behaviour in GSD as regards communication is different to that encountered in other environments (Cemile et al., 2009). The ability to persuade another and the willingness to cooperate decrease with distance, leading to a common deception of the team members (Bradner and Mark, 2002). It is therefore essential for students to develop the following set of skills (Monasor et al., 2010):

- Ability to communicate effectively using a common terminology and language.
- Performance in the use of synchronous and asynchronous means of communication.
- Informal communication and improvisation skills.
- Knowledge of language, cultural and ethical issues.
- Leadership and conflict resolution skills.
- Time management skills.
- Managing ambiguity and uncertainty. Ability to evaluate information critically.
- Skills to gain the interlocutor’s confidence and trust.
- Knowledge of negotiation skills and contract writing in a common language.
- Collaborative work skills.

However, it is difficult for educators to prepare students in these sorts of skills since they have to manage distributed activities in collaboration with distant institutions. Moreover, it is not easy to reproduce the conditions of real GSD developments, principally because of the resources required and the time limitations of the courses.

We present a framework whose objective is to provide theoretical and practical lessons that will

J. Monsor M., Vizcaíno A. and Piattini M..
A FRAMEWORK FOR DEFINING SIMULATORS WITH WHICH TO TRAIN GLOBAL SOFTWARE DEVELOPMENT.
DOI: 10.5220/0003459902610264
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 261-264
ISBN: 978-989-8425-77-5

allow students to acquire communicative and teamwork abilities through the simulation of multicultural GSD environments.

In order to provide support for these activities, we have developed VENTURE (Virtual ENvironment for Training cUlture and language problems in global softwaRe dEvelopment), a virtual training environment that places students in a simulated GSD scenario in which they interact with Virtual Agents (VAs) from different cultures that simulate realistic experiences. These simulations are carried out by using written communication tools, such as e-mail and instant messaging.

2 VENTURE

VENTURE consists of a platform integrated into an e-learning system that supports the GSD educational framework presented.

The innovative value of this framework lies in its rigorous support for training, in that it not only copes with cultural and linguistic differences, but also improves attitudes for collaborative group work without the need for real partners. The aim is to provide theoretical lessons and simulated practices in GSD, supported by a tool that simulates realistic GSD collaborative environments. Teachers provide students with the theoretical lessons on GSD activities in class, and students can then go on to use VENTURE to execute the training scenarios in the laboratory.

VENTURE simulates GSD virtual meetings in which students interact with VAs using a common language (usually English). VAs communicate with students textually and in an autonomous manner in order to allow them to confront cultural and linguistic problems similar to those that appear in real environments.

Learners are placed in a virtual scenario and work on the resolution of certain GSD activities by interacting with VAs of different cultures. Students can play the different roles in the process of GSD by interacting with the VAs. The interactions are guided by a Virtual Colleague (VC), which is another kind of VA that will also correct the students' inappropriate interventions.

VENTURE introduces students to the characteristics of the project to be developed and their role. They are also given a software engineering task or a responsibility that they must accomplish.

Each lesson has practical materials associated with it that must be delivered at prescribed milestones. The students will generally have to complete a document, or develop software in accordance with the purpose of the scenario. They must also answer a questionnaire at the end of the course.

The virtual meetings are designed to reflect the typical problematic or controversial situations encountered in GSD, and the students are therefore encouraged to find a solution to the problems by interacting in the correct manner.

The definition of these meetings is based on VTRML (VenTuRe Markup Language), which is an extension to the XAML (Extensible Application Markup Language) and permits the definition of all the elements required, including the cultural and linguistic rules for the scenario. The principal objective of this is to improve the students' skills and performance when dealing with cultural problems during the simulated conversations.

3 ARCHITECTURE OF VENTURE

VENTURE permits the design and development of GSD simulators by means of a client-server architecture which is presented in Figure 1.

On the server side, an *e-learning application* provides students with the course resources and the facilities needed to carry out the training activities. This part is made up of the following modules:

Resource Repository (1)**:** This contains the theoretical lessons assigned. Each theoretical lesson is formed of virtual meetings which can be executed at any time.

Task Area (2): A repository including the practical activities assigned, in which the delivery deadlines are specified. In this area, it is also possible for the students to upload deliverables and review the evaluation and instructor's comments for these activities.

Forum and Wiki Module (3, 4): This is used by the instructors to include global communications, and by the students to interact with instructors and partners.

Evaluation Area (5): Continual exams and questionnaires which serve to evaluate the students' learning.

On the server side, the theoretical material is stored in the *Pedagogical Module* (6) and is structured with reference to the knowledge areas: software requirements, software design, software construction, software testing, software quality,

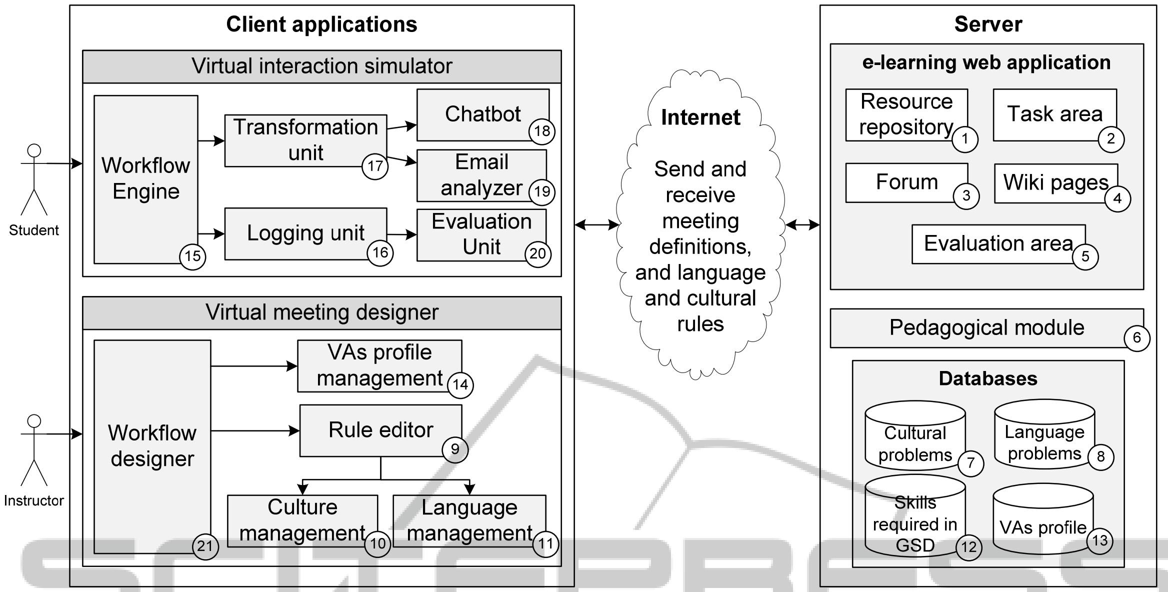

Figure 1: Venture architecture.

software maintenance, configuration management, software engineering management and software engineering process.

The *cultural problems* database (7) consists of a repository that contains the set of cultural problems and recognized differences that might affect communication in GSD scenarios. The linguistic problems that can appear when participants interact textually with a non-native language are stored in the *language problems* database (8). The linguistic rules considered are classified according to the kind of problem that they deal with and include any relevant information that may be useful for correcting the students' actions. The information contained in both the *cultural database* and the *linguistic database* is managed by the *Rules Editor* interface (9), which is made available to the instructors through its *cultural management* module (10) and *language management* module (11).

The *skills required in GSD* are stored in the database (12) which contains best practices for training the skills needed in GSD.

The *VA profile (13)* can be managed through the *VA profile management* module (14) which permits new characters to be included or existing ones to be modified, and also allows them to be incorporated into the theoretical materials or practical scenarios.

The *Workflow Engine* (15) is responsible for executing the meeting workflows by reading the definition of the meeting, and orchestrating the sequential execution of the corresponding phases.

The engine additionally makes it possible to save the log through the *login unit* (16) of the conversation, so that the instructor can review it later.

This workflow engine reads all the information related to the simulation and makes the appropriate transformations for each phase in order to generate the AIML language. It does this by using the *transformation unit* (17) to obtain information that is understandable by the *chatbot system* (18), in the case of synchronous interactions, and the *Email analyzer* (19), in the case of asynchronous interactions. The *Evaluation unit* (20) serves as a tracking mechanism of the students actions to inform both students and instructors of their skills and results from the use of this framework, providing a continuous and real-time evaluation.

The *Workflow Designer* (21) is a graphical tool that the course designers use to define and modify the virtual meetings. The virtual meetings are structured as sequential workflows made up of a set of phases containing the specifics details of the conversation for that phase.

4 CONCLUSIONS AND RESEARCH AGENDA

The chief advantage of using VENTURE is that students are more independent as they do not need to interact with distant partners and they can train at any moment without depending on the availability of other partners or colleagues. Students can also play different roles while interacting with VAs and learn about the different kinds of problems that may occur from different perspectives. Furthermore, since the VA controls the conversations it is not likely that off-topic conversations will occur, which quite often takes place when two or more students communicate by chat. Students can therefore take better advantage of their time than when working with other students.

In our future work we will test the framework presented by collecting evidence that will allow its effectiveness to be demonstrated.

We will guide our future efforts towards answering the following research questions:

- *Adequacy*: Do the students understand the purpose of the simulations? Do the students feel that they have improved their skills in GSD?
- *Time Requirements*: How long do the students need to complete the course and to finish the deliverables? Do instructors appreciate a faster organization and better performance in the GSD courses?
- *Usability*: What problems occur during the interaction with VAs when using VENTURE? What is the students' opinion of the feasibility of the tool?
- *Motivation*: Do students feel motivated when interacting with VAs? To what extent does a student perceive the usefulness of the framework?

With regard to the use of VAs, we intend to study to what degree they induce a sense of social presence in the students. This sense is increased by the transmission of nonverbal cues showing emotional states and gestures. In this respect, we shall also study the following questions:

- Do students identify with their roles in the virtual simulations?
- Do they consider that the experience of interacting with VAs is realistic?

From the perspective of the instructors, we must also study the feasibility of the framework for designing and customizing the training materials and scenarios. In this respect, the following questions should be answered:

- What is the instructors' opinion as regards the use of the meeting designer? How long does it take them to design a scenario?
- *Evaluation and monitoring*: What is their perception of the monitoring and evaluation facilities?

The students' active participation in the evaluation will be helpful in obtaining feedback with which to improve the framework. We are also in the process of preparing surveys, structured interviews and in-situ observations.

Finally, we also wish to design and test practical scenarios that will allow us to compare the performance of students who have trained with our framework with that of others of the same characteristics who have not.

The results of the evaluations will eventually help to design new training scenarios and to improve the cultural and linguistic problems database.

ACKNOWLEDGEMENTS

This work has been funded by the PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondos FEDER, TIN2009-13718-C02-01). It is also supported by MEVALHE (HITO-09-126) and ENGLOBAS (PII2I09-0147-8235), funded by Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), and co-funded by Fondos FEDER, as well as as well as GlOBALIA (PEII11-0291-5274), Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia in Spain, and ORIGIN (IDI-2010043 (1-5)) funded by CDTI and FEDER.

REFERENCES

Bradner, E. & Mark, G. 2002. Why distance matters: effects on cooperation, persuasion and deception. *Proceedings of the 2002 ACM conference on Computer supported cooperative work.* New Orleans, Louisiana, USA: ACM.

Casey, V. & Richardson, I. 2008. The Impact of Fear on the Operation of Virtual Teams. *Proceedings of the 2008 IEEE International Conference on Global Software Engineering.* IEEE Computer Society.

Cemile, S., Kathleen, S., Ferda, A., Robert, B., George, D. & Lopez, V. 2009. Exploring the communication behavior among global software development learners. *International Journal of Computer Applications in Technology (Accepted)*.

Monasor, M. J., Piattini, M. & Vizcaíno, A. 2009. Challenges and Improvements in Distributed Software Development: A Systematic Review. *Advances in Software Engineering,* 2009**,** 14.

Monasor, M. J., Vizcaíno, A., Piattini, M. & Caballero, I. Year. Preparing students and engineers for Global *Software Development: A Systematic Review. In: International Conference on Global Software Development (ICGSE 2010), August 23-26, 2010 2010 Princeton, NJ, USA. 177-186.*