JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE
J. Softw. Maint. Evol.: Res. Pract. 2010; **00**:1–5
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/smr

Cultural and Linguistic Problems in GSD: A Simulator to Train Engineers in These Issues

Miguel J. Monasor, Aurora Vizcaíno, 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
MiguelJ.Monasor@gmail.com
{Aurora.Vizcaino, Mario.Piattini}@uclm.es

SUMMARY

This paper tackles the need for universities and practitioners to train their students and engineers in the new challenges that Global Software Development (GSD) entails, which are principally related to communication, collaboration and cultural differences. Teaching the necessary skills requires practical experience. However, training in educational environments is difficult, and some challenging issues must still be confronted.

We have focused our work on the development of a virtual training environment that can simulate GSD scenarios involving Virtual Agents (VAs) from different cultures. The VAs interact with learners, who use typical communication tools to solve predefined problems. This environment considers common problems caused by distance and cultural differences when using English as a means of communication. It allows learners to train at any time, since the VAs are always available, and it also permits them to play different roles in the various stages of the project. In this paper we depict the design and development of the tool, as well as an initial evaluation.

KEY WORDS: Global Software Development, Engineering Education, Educational Environment, Teaching Model, Virtual Agents.

1. INTRODUCTION

The need to train both students and software engineers in the problems that they will confront in Global Software Development (GSD) environments has led to the necessity

for Software Engineering Education to be adapted in order to tackle the new challenges that GSD involves [1].

The main drawbacks of this type of development are the result of distance, time and cultural and linguistic differences [2]. Communication and coordination become more complex, since traditional face-to-face meetings are not the rule in distributed projects. Members from different countries interact by using their cultural knowledge and communication styles [3], thus interpreting communication from their particular perspective [4]. Interaction in GSD usually occurs through the use of communication tools when messages are sent or received. Learners must know how to adapt these messages so that the interlocutor can understand them without missing information or creating a misunderstanding [4]. In order to achieve this, participants must be trained in the specific cultural problems of GSD. However, this training is not usually part of conventional education [5]. Furthermore, training learners in GSD activities is a complex task that requires new theoretical contents, training environments and practical experiences, which commonly entails a great deal of coordination with distant members and institutions and is not always practicable.

In this paper we present a concept for an initial implementation of VENTURE, a virtual training environment that places learners in a simulated GSD scenario in which they are involved in realistic experiences through their interaction with Virtual Agents (VAs) from different cultures. The use of VAs has already proved to be useful in the training of multicultural communication [3]. Furthermore, they have the advantage that they are always available, signifying that users can work with this tool at any moment.

VENTURE has two basic objectives: to promote the acquisition of communicative and teamwork abilities in GSD environments and to provide support for a training framework that can be adjusted to train different cultures. Through this paper we depict the design and development of VENTURE, and we provide the results of an initial evaluation of the tool. A research agenda is also presented showing the main focus of our future work.

This paper is organized as follows: Section 2 explains the cultural and linguistic problems involved in GSD. Section 3 describes works related to the teaching of GSD. Section 4 describes VENTURE, and details of how this trains people in cultural and linguistic problems are provided in Section 5, while an example scenario is shown in Section 6. Finally, Section 7 shows a pre-evaluation of the environment, Section 8 explains our future work and Section 9 provides some concluding remarks.

DOI: 10.1002/smr

2. CULTURAL AND LINGUISTIC PROBLEMS

Some representative problems that occur in GSD are briefly presented below:

- Participants do not speak during team discussions until invited to do so, since this is not considered polite in their culture [6].
- The use of direct or indirect style refers to the way in which people reveal their intentions. Being too direct may appear very rude in some cultures [7].
- Succinct/elaborate style refers to the amount of information provided. Some cultures have problems in understanding silences or succinct messages [7].
- People from some cultures tend to say that they have understood something when they have not, thus causing problems that may be detected too late [6].
- Participants must know when to use formal or informal communication, how to do so, and when to use different communicative styles depending on the situation [7].
- The 'Mum Effect' [8]. This refers to the tendency to cover up critical information or to distort negative news by presenting it as more positive information.

A recent model dealing with cultural differences, presented by House [4], defines a set of cultural dimensions. Part of the aim of our framework consists of establishing a broad set of cultural problems that may appear in GSD, and measuring the degree to which they are affected by the different cultural dimensions set out by House.

In relation to the language barrier, the use of a non-native language entails particular problems such as the overuse of certain verbs of a high semantic generality (do, have, make, put, etc.), along with the use of false friends (when a word looks or sounds similar in two different languages, but differs in meaning and may cause misunderstandings) and grammatical inaccuracies that may lead to misunderstandings.

Typical recommendations when communicating with people from different cultures are [7], [6]: formulate criticism/praise carefully; use simple language, avoid slang, colloquialisms, jargon, acronyms and metaphors; avoid jokes that may be misinterpreted; and know how to alternate between informal/formal writing.

3. LITERATURE REVIEW

The following sections summarize a rigorous literature review [9] that was performed in order to answer the following research question: *What initiatives have been carried out in relation to Global Software Development training and education?*

DOI: 10.1002/smr

3.1. Learning environments

Experimental and research approaches were found. One example of this is iBistro [10], an augmented space based on the 'learning by doing' approach which enables distributed members to collaborate in the development of software.

[11] describes a collaborative environment for teaching distributed teams through the use of a set of tools (including chat, a scribble tool, an application sharing tool, etc.). This study also presents a Web portal that helps students to manage the groups and projects in which they are involved. In this category we also found ClockingIT [12], a project management and collaboration customizable platform that provides chat rooms, instant messaging, a built in wiki and discussion boards.

[13] explores the interaction with avatar-based humans through the integration of Second Life in virtual collaborative projects, which allows collaboration skills and intercultural competence to be trained by performing real-time realistic activities. In a similar vein, [14] present Teamlink, a Collaborative Virtual Environment based on configurable avatars in a virtual 3D world which supports icebreaking activities with the aim of establishing trust between virtual team members.

3.2. Academic training

Universities that teach GSD usually organize joint student developments to represent scenarios like those that can be found in industry, in which the students communicate using typical GSD tools such as email, telephone, instant messaging, wiki pages, forums and blogs. A representative example of this can be found in [5], which presents a European Masters program on Global Software Engineering.

[15] describes an experience in which distant team members collaborated by documenting each task in a project and sending it electronically to the other university groups in charge of developing them. We also found other courses in the field of collaborative learning, which is closely related to the teaching of GSD [16, 17].

A similar idea is reported under the name of Open Ended Group Projects (OEGPs) which is a supplement to conventional teaching approaches that can be used to develop the essential skills needed in the 'real world' [18]. One noteworthy experience is that of the OEGP Framework presented in [19], which provides a flexible educational course that evolves with the assistance of an action research program.

Finally, we should mention RUNESTONE [20], an International Student Collaboration initiative which consisted of a project-centred course in computer systems during which students were involved in the development of a pedagogical project involving two universities in different countries.

DOI: 10.1002/smr

3.3. Implications and deficiencies

The results found in the literature review allow us to affirm that reproducing GSD environments in educational contexts is difficult. Primary studies generally focus on improving the participants' collaborative skills and are oriented towards a specific stage of the software development [10].

One of the most important subjects consists of improving formal and informal communication skills, which is not part of traditional software engineering education at this time. In this respect, we found theoretical classes based on the challenges of cultural diversity. However, this is not efficient without practice with real partners from other cultures. Some studies discuss the interaction of participants in a distributed network [5] or through the use of avatar-based systems [13]. However, the participants' different timetables make it difficult to coordinate the training projects and may even damage relationships between participants. Moreover, this means of training people in cultural problems does not provide opportunities to make corrections, since teachers cannot attend all of the meetings and provide feedback.

In this work we outline a simulation proposal oriented towards training the specific problems of GSD with the aim of diminishing the problems of the traditional methods.

4. A VIRTUAL TRAINING ENVIRONMENT

VENTURE consists of an architecture based on an e-learning platform which provides support to a complete GSD training framework focused on the training of cultural and linguistic differences and also on improving collaborative group work attitudes in the context of GSD without requiring real partners.

In the context of our research, we have defined a GSD training environment that is made up of a set of plugins that interact with the e-learning platform and which are oriented towards the simulation of context-sensitive interactions with the different stakeholders who may participate in GSD. These software plugins are focused on the training of specific activities that follow a workflow which is defined by metadata detailing the contextual information required for the simulation.

The system is based on the Problem Based Learning approach in which learners work on the resolution of a problem in a virtual training scenario where they must perform some typical GSD activities by interacting with VAs from different cultures who will play a role in the scenario. The learners must learn how to interact with the VAs, and will confront problems specially designed to provide training in cultural and linguistic problems depending on the participants' cultures.

DOI: 10.1002/smr

4.1. Description of the Architecture

As is shown in Figure 1, the core of our environment is the *e-learning site*, with which we define our training courses including a set of predefined elements such as: *wiki pages*, *forums*, *news*, *artefacts repository*, etc., which learners use to manage their activities and to collaborate with instructors or with other participants.

Each course can also contain virtual meetings, which are supported by means of the GSD plugins and are designed to help learners to improve their communicative and collaborative skills when using a common language (generally English) by interacting with VAs from different cultures. We have developed two *GSD plugins*:

Chat plugin: A chat consists of one or more VAs that can participate by playing a specific role in the GSD project (e.g. customer, requirements analyst, developer, project manager, etc.). The virtual meeting is guided by a Virtual Colleague (VC), a special VA designed to help learners to cope with the meeting simulation. The VC guides the learners and corrects their actions, providing rationales and explaining their consequences, along with providing opportunities for self-reflection and self-correction. The conversation is time-limited and the learner must obtain all the relevant information in an appropriate manner, whilst taking into account the VAs' culture.

E-mail plugin: In this case, when the VA receives the learner's email, it is analyzed and evaluated by searching for cultural and linguistic mistakes, and also by considering the structure of the text and the organization of the ideas based on the analysis of the keywords that were expected in each paragraph.

The *application data* contains the information concerning the application management and information about the instructors and learners that can be used to manage their accounts, history and evaluations. The *Virtual Agents database* contains the information related to the VAs' appearance, gestures and voices.

The *cultural knowledge* and the *language knowledge* databases contain the rules which are classified by the type of problem that they intend to tackle, and are used by our training plugins to correct the learner's interaction. These rules are also categorized according to the different pairs of cultures and languages that might participate in a meeting, thus permitting the design of scenarios in different languages or dialects. Finally, the *training scenarios database* contains the GSD training scenario definitions with all the information required for their execution. This information is accessible through the *Server Services*, which allows *GSD plugins* to access the required information in order to perform the simulation.

A GSD training scenario (based on either chat or e-mail) consists of a set of phases that define a small part of the interaction, thus making up a complete *Meeting Workflow*

DOI: 10.1002/smr

that establishes the flow of the conversation. Each phase of the workflow is defined by specific conversational knowledge and the language and cultural knowledge which are used in the context of that phase in the conversation. This knowledge is stored as a set of text patterns that are interpreted by means of a chatbot system.

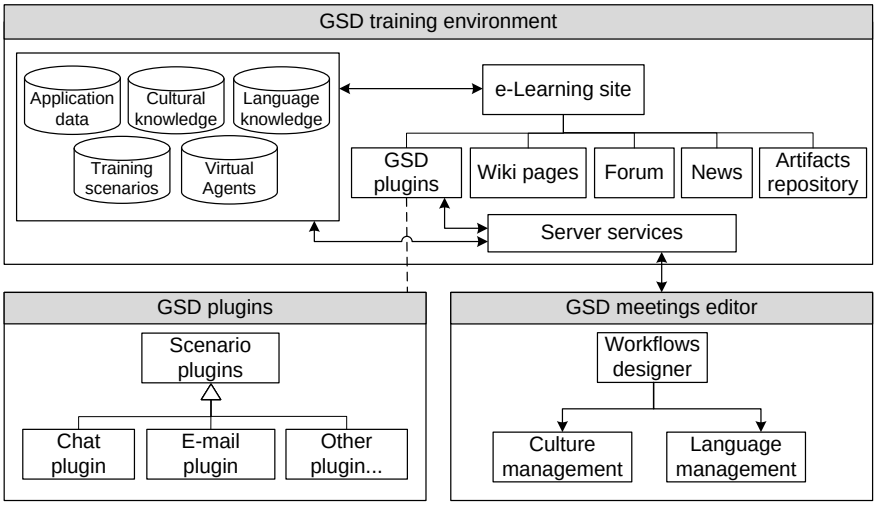

Figure 1. GSD training environment architecture

By considering this architecture, an **instructor** can manage both the training courses and the learner's activities. In this respect, the system provides "informating up" mechanisms [21], since instructors can know the status of the tasks in real time.

One important element of the environment for the instructors is the *Workflows designer*, with which they can design the virtual interactions which are defined by a set of phases. The designer allows the complete sequential workflow to be designed by including the phases that will participate in the interaction. Each phase must be defined at a high granularity level, containing the information for a specific interaction containing the following contextual information which is embedded as metadata (see example in Figure 3):

- The *conversational knowledge* required for the VAs to answer the learner's questions. This is defined by using AIML (Artificial Intelligence Markup Language), a pattern language interpreted by the chatbot engine in which: a *pattern* is a string of characters intended to match one or more user inputs, and a *template* specifies the response to a matched pattern.
- The *cultural* and *linguistic rules*: Our system uses a pattern with the incorrect interactions expected from the users and a template with the response that the VC

DOI: 10.1002/smr

will provide to correct them and provide feedback. These rules can be imported from the aforementioned *cultural* and *language knowledge bases*. Instructors can therefore manage the language and cultural knowledge bases through the *culture management* and *language management* applications, with which they can add new rules or edit the existing ones.

- Additional metadata: Other associated information, including the VAs' emotions and gestures, type of cultural or language problem and its severity, artefacts that the user could use or complete in order to carry out a task.

From the **learner's** point of view, our system provides "informating down" facilities [21], since users can create and share knowledge with their peers and instructors. When they log into the system, they can begin one of the assigned training scenarios. They first receive a theoretical background and they then practice what they have learned about the problems using the *GSD plugins*. Learners have information about their performance, deadlines, qualifications, historical actions and instructors' comments at their disposal. They also have a shared repository containing any deliverables required to solve the scenario, such as documents and associated diagrams. Learners can also know which team members are online and subscribe to automatic notifications about news or changes in the document repository, or fill out questionnaires and take exams.

5. PROVIDING TRAINING FOR CULTURAL AND LINGUISTIC PROBLEMS

In this section we explain how VENTURE detects cultural and linguistic problems during the interaction. The system stores the cultural and linguistic rules in a central database so that this knowledge can be reused.

5.1. Cultural Problems

The cultural knowledge stored in the *cultural database* contains the cultural problems, introduced in Section 2, which may appear in GSD. For each cultural problem that may appear when two different cultures interact, the system measures the degree to which this is affected by the cultural dimensions laid out by House and considers this in order to activate or deactivate certain rules depending on the learner's and the VA's cultures.

However, each specific cultural problem requires different techniques to be worked on or detected. For example, when participants do not speak during team discussions until being invited to do so, this can be detected by including a time limit in the phase for the participant responses during which they must interact. Other problems such as

DOI: 10.1002/smr

the correct use of formal or informal communication are addressed by detecting certain keywords in the conversation that are typical of each communication style.

5.2. Linguistic Problems

In this case, the language database contains rules for the language scope which are defined in AIML format. These rules contemplate issues such as the overuse of certain verbs of a high semantic generality and the use of false friends. We also correct grammatical inaccuracies by using third party dictionaries and grammatical correctors for the target language, thus signifying that the VC is able to report errors such as the avoidance of passive forms, incorrect plural formations, the absence of the third person, the use of redundant prepositions, etc.

6. EXAMPLE OF A TRAINING SCENARIO

In this section we present a training scenario for the Requirements Elicitation (RE) activity, since this is a highly communicative process which is particularly affected by poor communication along with the cultural and linguistic differences.
With this aim in mind, we developed a theoretical course on RE focusing on the problems that Spanish learners face when interacting using English in GSD environments. Users, therefore, access to the e-learning system and study the theoretical material. After each lesson, they usually have available a virtual meeting that they can execute in order to train the concepts of that specific lesson. In this case, they play the role of analysts in order to elicit a set of requirements for a virtual customer from the United States.

The example in Figure 2 shows a chat conversation in which our system trains participants with regard to the Mum Effect problem. In this scenario the customer asks the learner to submit a deliverable for tomorrow. But s/he does not have time to do so, and must communicate this to the customer rather than attempting to cover up or minimize the problem. In the example, we show how the learner tries to avoid conveying the bad news, and how the VC detects this and advises the learner to be more precise in his/her response. After communicating the bad news, the learner must attempt to extend the deadline by discussing the problem with the VA in a maximum of 25 minutes. We also show how the learner makes a language mistake and how the VC corrects him/her. The characters were provided by Haptek Incorporated (http://www.haptek.com), and include gestures and text-to-speech features, which help

DOI: 10.1002/smr

to motivate the learners and to teach them accents and typical gestures that can be misinterpreted. These gestures are also defined as metadata along with the conversational knowledge.

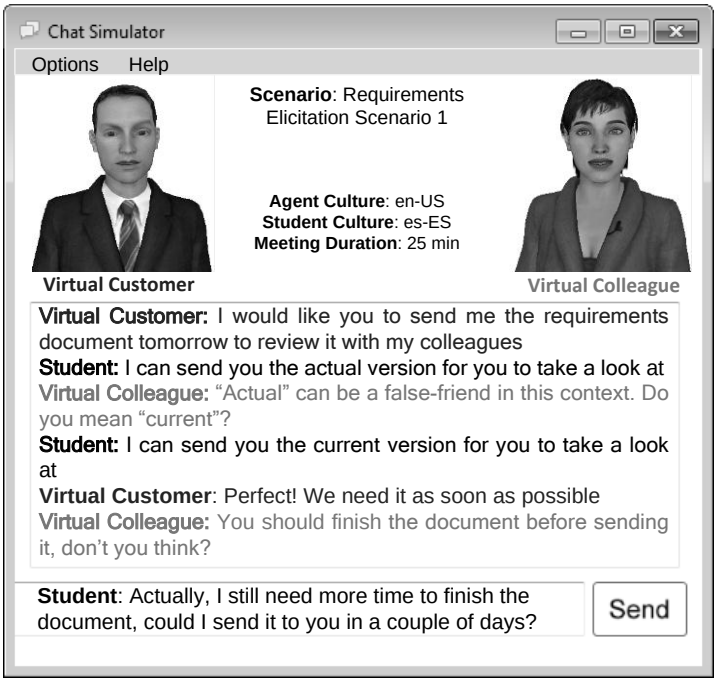

Figure 2. Example of virtual meeting

In Figure 3, we show a fragment of the Meeting Workflow corresponding to this scenario and detail the phase that corresponds with the conversation shown in Figure 2. The workflow starts with the "Introduction" phase, in which the VC will introduce the participants and present the VA. This initial phase is completed when the learner greets the VA, and the "Document request" phase then begins automatically.

We have detailed the content of this phase, including the VA's conversational knowledge and the cultural and linguistic rules for this context. For each learner intervention, our system reviews the text introduced and checks the patterns defined for cultural and linguistic problems.

With regard to conversational knowledge, the phase begins with the participation of the customer, who is requesting something from the learner. The learner must then interact by asking what the customer expects from him in order to provide an answer

DOI: 10.1002/smr

based on the AIML definition. In this case the customer is expecting a *negative sentence* in order to provide an answer that will finish the phase to allow the next one to begin.

If the learner provides a *positive sentence*, the phase includes a cultural rule concerning the Mum Effect which is triggered in order to correct this behaviour. In this case the VC encourages the learner to answer with a negative sentence in an attempt to postpone the delivery of the document. With regard to language knowledge, this phase also detects the pattern "actual", which is classified as a false friend mistake that the VC will correct.

Note that in this example we use the simplification "negative sentence" and "positive sentence" in order to avoid representing all the positive or negative patterns that the system actually considers. Examples of negative patterns in this context are: *can't *, *impossible*, *difficult*, etc., and the sentence is considered to be positive when none of these patterns are matched.

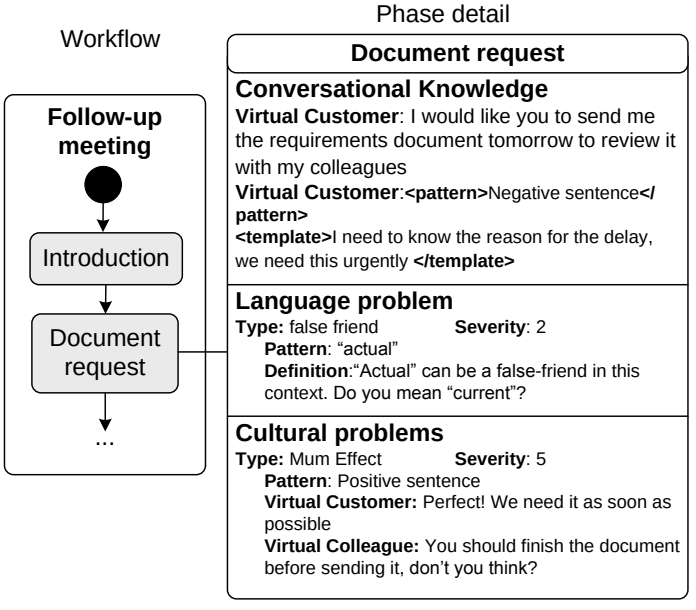

Figure 3. Definition of virtual scenario

At the end of a virtual meeting, the learner has available, through the e-learning system a set of exercises, which may consists of filling in a document or answer a questionnaire to check their understanding of the practical activity as well as the theoretical lesson.

DOI: 10.1002/smr

7. PRE-EVALUATION

An evaluation of VENTURE was carried out by 4 members of 3 multinational companies. Two of these were project manager, whilst the others were developers, and all of them had experience in GSD. The process consisted of a textual and verbal presentation of the environment and an interview including questions related to their perception with regard to the use of the environment in their companies, its usefulness, and its usability from the students' and instructors' perspectives.

The first finding was that all of them considered VENTURE to be innovative in comparison with their classic online learning systems. They all also agreed that it could be useful in their companies and that are interested in using it, although they remarked that the time available for training in their companies is quite limited. This helped us to confirm that a duration of 30 minutes or less for each lesson would be appropriate..

Two responders stated that one of the main problems that companies usually encounter when organizing courses is related to the difficulty involved in finding experts on this subject, the time needed to develop these courses, and gathering together sufficient participants to carry them out. All of the above reasons led them to state that they believed that the use of a simulator was an excellent idea, since all of these problems might be avoided. Moreover, all of them considered that the possibility of presenting controlled situations which would be difficult to systematically reproduce with real members was the system's strongest point. The flexibility that would make it possible to improve the training scenarios was also well valued, although they showed concern about the time that these improvements would require.

The interviewees did not mention any further needs from the point of view of the instructor. From the student's point, they found the chat plugin easy to use and motivating. However, they provided some tips by which to improve its usability, such as separating the VC's comments in order not to mix them with the conversation, including the option of being able to pause the interactions, and the possibility of interacting in a guided manner with real members rather than with VAs.

8. RESEARCH AGENDA

Our objective in the future is to collect evidence that will allow us to examine in what types of tasks VENTURE is most effective, what factors contribute to its usability and how the training scenarios must be improved to better the students' performance [22].

DOI: 10.1002/smr

The following research questions are intended to provide insights into how users cope with the environment, and may highlight possible problems:

- *Perceived usability*: What is the students' opinion of the feasibility of using VENTURE?, What problems occur when interacting with VAs using VENTURE?
- *Motivation*: Do students show higher motivation when interacting with VAs than when interacting with real partners?
- *Time requirements*: How long do the students need to complete the course?, Is the virtual meetings duration adecuate for accomplishing their scope?
- *Adapted to reality*: Do users think that the problems shown are similar to those found in reality?
- *Adequacy*: Do the students understand the contents and purpose of the training scenarios? What is their perception in relation to the training of the cultural and language problems? Do the students feel that they have improved their cultural and language skills?

The usability from the perspective of the instructors must be studied through the following questions:

- *Scenarios design*: How long do the instructors need to design a scenario?, What is their opinion with regard to the use of the workflow designer?
- *Evaluation facility*: What is the instructors' perception of the monitoring and evaluation facilities?

Students will be invited to participate in the preliminary evaluation, which we are currently designing, since we believe that it is better to obtain a rapid feedback in order to improve the principal deficiencies of the tool. A point here is how to cope with the asynchronous interactions. Since in real environments, participants have to wait for the reception of the responses, we still are not sure about the convenience of simulating realistic response times in our system, since it could be discouraging for the learners.

After the clarification of these points, we would obtain an evaluation from engineers who work in three companies that often collaborate with us on different projects. We are therefore preparing surveys, structured interviews and in situ observations.

9. CONCLUSIONS

In this paper we have presented the design and initial implementation of VENTURE; an environment that provides support for the training of GSD activities focusing particularly on cultural and linguistic diversity. An initial evaluation of the environment has also been presented along with a future research agenda.

DOI: 10.1002/smr

Learners can use this environment to play different roles in the software lifecycle by interacting with VAs which guide the conversation, and it allows them to train at any time, without depending on the availability of other partners or colleagues. The users therefore learn about the different kinds of problems that may occur from different perspectives. Furthermore, we avoid the inconvenience of involving non-expert participants in real projects, thus reducing the effort required to coordinate distant learners or institutions, and minimizing the instructors' effort and the costs of infrastructure and maintenance.

Since it is not possible for instructors to cover all the stages and problems of the GSD, our aim in our future work is to design a large set of training scenarios oriented towards improving communication, coordination and collaboration in the software requirements, software design, software construction, software testing, documentation and project management stages. Our simulator also permits the customization of these training scenarios, thus signifying that instructors can adapt them to their academic needs and enterprises can adapt them to their processes.

ACKNOWLEDGEMENT

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 ENGLOBAS (PII2I09-0147-8235), funded by the Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), and co-funded by Fondos FEDER and ORIGIN (IDI-2010043 (1-5)) funded by CDTI and FEDER, as well as GlOBALIA (PEII11-0291-5274), Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha.

REFERENCES

1. Monasor, M.J., M. Piattini, and A. Vizcaíno, *Challenges and Improvements in Distributed Software Development: A Systematic Review.* Advances in Software Engineering, 2009. **2009**: p. 1-16.
2. Damian, D., A. Hadwin, and B. Al-Ani, *Instructional design and assessment strategies for teaching global software development: a framework*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 685-690.
3. Endrass, B., E. André, L. Huang, and J. Gratch, *A data-driven approach to model culture-specific communication management styles for virtual agents*, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*. 2010, International Foundation for Autonomous Agents and Multiagent Systems: Toronto, Canada. p. 99-108.
4. House, R.J., et al., *Culture, Leadership, and Organizations: The GLOBE Study of 62 Societies*. 2004: Sage Publications. 848

DOI: 10.1002/smr

5. Lago, P., et al., *Towards a European Master Programme on Global Software Engineering*, in *20th Conference on Software Engineering Education & Training*. 2007, IEEE C. S. p. 184-194.
6. Anawati, D. and A. Craig, *Behavioral Adaptation Within Cross-Cultural Virtual Teams.* IEEE Transactions of professional communicacion, 2006. **49**(1): p. 44-56.
7. Shachaf, P., *Cultural diversity and information and communication technology impacts on global virtual teams: An exploratory study.* Inf. Manage., 2008. **45**(2): p. 131-142.
8. Ramingwong, S. and A.S.M. Sajeev, *A Multidimensional Model for Mum Effect in Offshore Outsourcing*, in *Proceedings of the 2008 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*. 2008, IEEE Computer Society. p. 237-240.
9. Monasor, M.J., A. Vizcaíno, M. Piattini, and I. Caballero. *Preparing students and engineers for Global Software Development: A Systematic Review*. in *International Conference on Global Software Development (ICGSE 2010)*. 2010. Princeton, NJ, USA. p. 177-186.
10. Braun, A., A.H. Dutoit, A.G. Harrer, and B. Brüge, *iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses*, in *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*. 2002, IEEE Computer Society. p. 197.
11. Swigger, K., et al. *Teaching Students How to Work in Global Software Development Environments*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006.* 2006. Atlanta, GA, USA.
12. Cajander, Å., et al. *Students analyzing their collaboration in an International Open Ended Group Project*. in *39th ASEE/IEEE Frontiers in Education Conference.* 2009. San Antonio, Texas IEEE
13. Corder, D. and A. U, *Integrating Second Life to enhance global intercultural collaboration projects.* ACM Inroads, 2010. **1**(3): p. 43-50.
14. Clear, T. and M. Daniels. *2D & 3D Introductory Processes in Virtual Groups*. in *33rd ASEE/IEEE Frontiers in Education Conference.* 2003. Boulder, Colorado: IEEE
15. Burnell, L.J., J.W. Priest, and J.R. Durrett, *Teaching Distributed Multidisciplinary Software Development.* IEEE Softw., 2002. **19**(5): p. 86-93.
16. Clear, T., *Exploring the notion of 'cultural fit' in global virtual collaborations.* ACM Inroads, 2010. **1**(3): p. 58-65.
17. Clear, T. and D. Kassabova, *A Course in Collaborative Computing: Collaborative Learning and Research with a Global Perspective*, in *Proceedings of the 39th ACM Technical Symposium on Computer Science Education*, M. Guzdial and S. Fitzgerald, Editors. 2008, Portland, OR. p. 63-67
18. Daniels, M., *The contribution of open ended group projects to international student collaborations.* ACM Inroads, 2010. **1**(3): p. 79-84.
19. Daniels, M., Å. Cajander, A. Pears, and T. Clear, *Engineering Education Research in Practice: Evolving Use of Open Ended Group Projects as a Pedagogical Strategy for Developing Skills in Global Collaboration (Special issue on Applications of Engineering Education Research).* International Journal of Engineering Education 2010. **26**(4): p. 795-806.
20. Daniels, M., et al. *RUNESTONE, an International Student Collaboration Project*. in *IEEE Frontiers in Education Conference*. 1998. Tempe, Arizona: IEEE
21. Leidner, D.E. and S.L. Jarvenpaa, *The use of information technology to enhance management school education: a theoretical view.* MIS Q., 1995. **19**(3): p. 265-291.
22. Almstrum, V.L., et al., *Evaluation: turning technology from toy to tool: report of the working group on evaluation*, in *Proceedings of the 1st conference on Integrating technology into computer science education*. 1996, ACM: Barcelona, Spain. p. 201-217.

DOI: 10.1002/smr