An Architecture for Creating Simulators for Training Global Software Development

Miguel J. Monasor
University of Castilla-La Mancha
Campus Universitario s/n, 02071, Albacete, Spain
MiguelJ.Monasor@gmail.com

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
{Aurora.Vizcaino, Mario.Piattini}@uclm.es

Abstract— **The need for universities to provide their students with theoretical lessons and practical training in GSD has led to the development of new training schemes. Most of these consist of involving distant students from different cultures in the development of software projects in order to allow them to confront similar problems to those that occur in real environments. This, however, is not always possible since it requires a great deal of coordination among various universities. We propose to avoid these problems by using a simulator-trainer with virtual agents which will permit users to work at any time. The virtual agents will induce users to confront the most classical problems in GSD. In order to construct this simulator, we propose an architecture which has different modules to support cultural, language and communication problems. This paper describes this architecture and how to use it, along with the means employed to evaluate it.**

Keywords-global software development; distributed software development; teaching; education; training; learning; simulator

I. INTRODUCTION

Universities are currently teaching Global Software Development (GSD) with the aim of training Software Engineering students in the challenges that it entails [1]. However, training students in GSD activities implies the development of specific skills that are not common in the traditional co-located environments, since the students must learn how to take part in a team by interacting with members from different cultures and different time zones in a common language.

Training these skills is difficult since it requires realistic practices that allow students to be placed in similar scenarios to those that occur in the real world. The techniques commonly applied by universities consist of involving distant students from universities in different countries in the joint development of a piece of software by imitating the GSD conditions in real business environments [2].

Students are provided with theoretical notions of the problems that they may encounter in GSD and are involved in experiences in which they have to overcome the challenges of real problems by collaborating with other students. They thus confront the communication drawbacks involved in interacting with a multicultural team in a common language. The participants are usually non native speakers, which modifies the teams' behaviour and performance and assists them to change their attitudes.

Non native speakers must therefore learn to communicate in a second language whilst developing interactive competences and negotiation skills [3], and native speakers must be aware of the problems that may appear in this respect. Students must learn develop the ability to cooperate, persuade, negotiate and generate trust among team members [4].

However, it is difficult for universities to prepare students in these sorts of courses since instructors 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, the time limitations of the courses or the different timetables or requirements of the universities involved. The aforementioned reasons signify that this training is not always possible and the need for new teaching methods that require less organizational efforts arises.

In this paper we present a framework for the provision of theoretical lessons and practical training through the simulation of GSD scenarios. These simulations will allow students to acquire communicative and teamwork abilities without the need to interact with real members since they will interact with Virtual Agents (VAs) which simulate that they are from different cultures and play a specific role in the GSD scenario. The advantage of using VAs is that they are always available, meaning that users can train at any time. We also use a Virtual Colleague (VC), which is a special VA that will guide the interactions and will correct the students and provide them with feedback. This is, therefore, like having a "brilliant" student helping you when it is necessary.

In order to provide support for these simulations, we have developed VENTURE (Virtual ENvironment for Training cUlture and language problems in global softwaRe dEvelopment), a platform for creating customized scenarios which are executed by placing students in simulated written GSD interactions with VAs using chat or email.

VENTURE provides an active participative learning environment in which students confront predefined experiences and obtain immediate feedback and reflections after the experience. These experiences can also be repeated, and all the students can therefore have common experiences which are not possible in real environments since each student has to assume a specific role. These experiences are oriented towards confronting specific situations depending on the students' skills and culture, thus allowing the definition of a broad set of problems and situations in addition to levels of difficulty. VENTURE also assesses the students' interventions through the simulations, which are defined in order to maximize the apparition of the typical problems encountered in GSD in a controlled manner.

VENTURE permits instructors to design virtual meetings and interviews using different avatars. The definition of the meeting is based on a workflow model which structures the information required for the interaction, and which is easily customizable to the different cultural and language problems that will be addressed during the interactions between customers, vendors, and colleagues. Finally, the design of a proposal with which to evaluate students' use of this framework is also presented.

This paper is organized as follows: Section II summarizes the related works in the field of GSD training and education. Section III describes the architecture of VENTURE, and an explanation of its use in training synchronous and asynchronous means of communication is shown in Section IV. Finally, Section V presents some concluding remarks and Section VI provides an outline of our future work.

II. Background and Motivation

In an initial stage of this research we carried out a systematic literature review in the field of GSD training and education, which we presented in the ICGSE 2010 [1]. This work led to the discovery that most of the studies reviewed attempt to reproduce scenarios like those found in industry. A representative example of this is presented in [2] and consists of a university course on which a certain number of lectures provided theoretical notions to be applied in GSD projects. The practical activities carried out involved two universities, and allowed cultural differences and communication and coordination issues to be trained. The students participated in a software project by following a waterfall model and interacted using collaborative and communicative tools. These kinds of programs are usually applied towards the end of the education study program and on Master's courses [5]. An example of this is a European Masters program on Global Software Engineering involving several universities from different countries which is presented in [6].

In the field of Open Ended Group Projects (OEGPs), similar teaching approaches also exist to develop the essential skills needed for collaboration and communication in multicultural environments [7]. In this respect, the experience presented in [8] provides a flexible educational course which evolves by means of the assistance of an action research program that allows gradual improvement through a combination of learning theories and stakeholder input.

In [9] the authors developed a GSD course by studying the challenges and solutions for task allocation, communication issues, time zone differences, communication and coordination complexity, infrastructure support, cultural issues and technical issues. These authors also present some teaching patterns by which to involve students in the problems related to GSD activities. These patterns are specifically intended to simulate real-world interactions in the stages of GSD, such as design, implementation, testing, and support.

RUNESTONE [10] was an international project that consisted of a project-centered course in computer systems in which students from different countries and with different backgrounds had to handle cross-cultural and linguistic problems in order to develop a project. A similar experience is also presented in [11].

Various educational approaches in enterprises also appear outside the academic field [12], [13]. In these cases, the concept of "learning networks" is commonly applied [14]. Learning networks are a means of teaching that take advantage of the knowledge of workers in a company who can train learners in specific skills in which they are experts.

A. *Tools and Collaborative Environments*

Those studies that refer to training in universities, commonly report the use of collaborative tools to support the training process. A popular example is iBistro [15], which is an augmented space in which distributed students collaborate in the development of a software project. It addresses miscommunications and information problems in informal meetings and helps to improve social skills. Similarly, the web-based collaborative platform presented in [16], allows instructors to manage training activities and students to work with their partners in order to achieve the module's scopes by using communication and collaborative tools.

A further example, ClockingIT [17], is an open-source customizable web-based platform that provides instant messaging features, task assignment and management, discussion boards, document sharing and wiki pages with which to share news and information about a project in a collaborative manner.

The use of collaborative environments is also reported in literature. Many of these studies deal with the use of avatars in the training of interaction skills as a means to confront students with typical cultural and communication problems. An example of this is reported in [18] in which the authors use avatar-based humans during the development of virtual collaborative projects. The avatars in this system can access a virtual room in which they can to walk, run and interact in real-time with other avatars by showing emotion and gestures.

Teamlink [19] is another collaborative virtual environment in which configurable avatars interact in a 3D virtual space. The purpose of Teamlink is to support icebreaking activities and promote the establishment of trust between virtual team members by using asynchronous means of communication. A similar approach is UniWorld [20].

Finally, we should mention, CURE [21], a virtual learning environment, which uses virtual rooms for collaboration among team members. These rooms may contain pages (content), communication channels (such as chat, mail, etc.), and users, who will interact with other users located in the same room.

B. Discussion

Reproducing GSD environments in educational contexts is difficult, and most studies report organizational problems when attempting to achieve close collaboration between students from different countries. Moreover, the students involved in these training activities frequently show little motivation, or possess different levels of knowledge or skills which makes it necessary to provide them with different training strategies [22]. Team members' cultural and language backgrounds must therefore be borne in mind in the design of the current courses, in order to obtain multicultural groups and ensure a realistic simulation of global virtual collaboration [23].

Another of the problems reported in literature is the ineffectiveness of communication through chat or email with other students, which leads to missed deadlines. These problems are also frequently related to technical issues [24].

Providing a rigorous training in communication and cultural issues is another subject which is not covered in the existing proposals. Since these problems may appear randomly during interactions, depending on the specific circumstances of certain settings [24], this subject must be dealt with in depth by using a rigorous method. The current methods also make it difficult to provide opportunities to make corrections, since teachers cannot attend all the meetings and provide feedback.

In summary, we consider that current teaching methods have the following limitations:

- Limitations in the number of people that can participate and problems in finding an appropriate multicultural group.
- Difficulties to establish relationships with other universities or institutions.
- Students depend on their colleagues' availability and skills.
- Students do not have common experiences. Everyone plays a different role and it is difficult to evaluate them fairly.
- It is difficult to provide opportunities for self-reflection and feedback. The repetition of tasks is not always possible under the same conditions.
- Lack of rigorous orientation towards cultural and communication difficulties.

III. ARCHITECTURE OF VENTURE

This section shows the main components of VENTURE, the platform that we have developed to create customized simulators. In order to clarify the function of some of these components we show an example of their application by using a simulator in which a Spanish student playing the role of requirements analyst will interact in English with a man from Russia (VA) who plays the role of client.

The architecture of VENTURE follows client-server structure, as is presented in Figure 1. An *e-learning application* is the core of the client side, which provides the course resources and the informating down facilities [25] by allowing the students to know their evolution and state. It also provides informating up facilities, since instructors can know, in real time, the status of the students' tasks and actions, and communicate with them. The *e-learning application* is made up of the following components:

Resource repository (1)**:** in which both the theoretical lessons and the simulators and artifacts are made available to the students.

Task area (2): which serves to control and schedule the practical activities. The students can also upload deliverables in this area.

Forum and wiki module (3, 4): through which students and instructors can keep in touch.

Evaluation area (5): in which students can do exams, fill in questionnaires, and review the evaluation and the instructor's comments for these activities.

A. Pedagogical module

The *Pedagogical Module* (6) stores all the theoretical contents in the field of teaching GSD training and is structured with reference to the following knowledge areas: software requirements, software design, software construction, software testing, software quality, software maintenance, configuration management, software engineering management and software engineering process.

The difference between this module and the *Resource Repository* is that the latter contains the general contents that are available to all the students. The *Pedagogical module* also contains the different strategies needed to train specific skills according to the needs of each student.

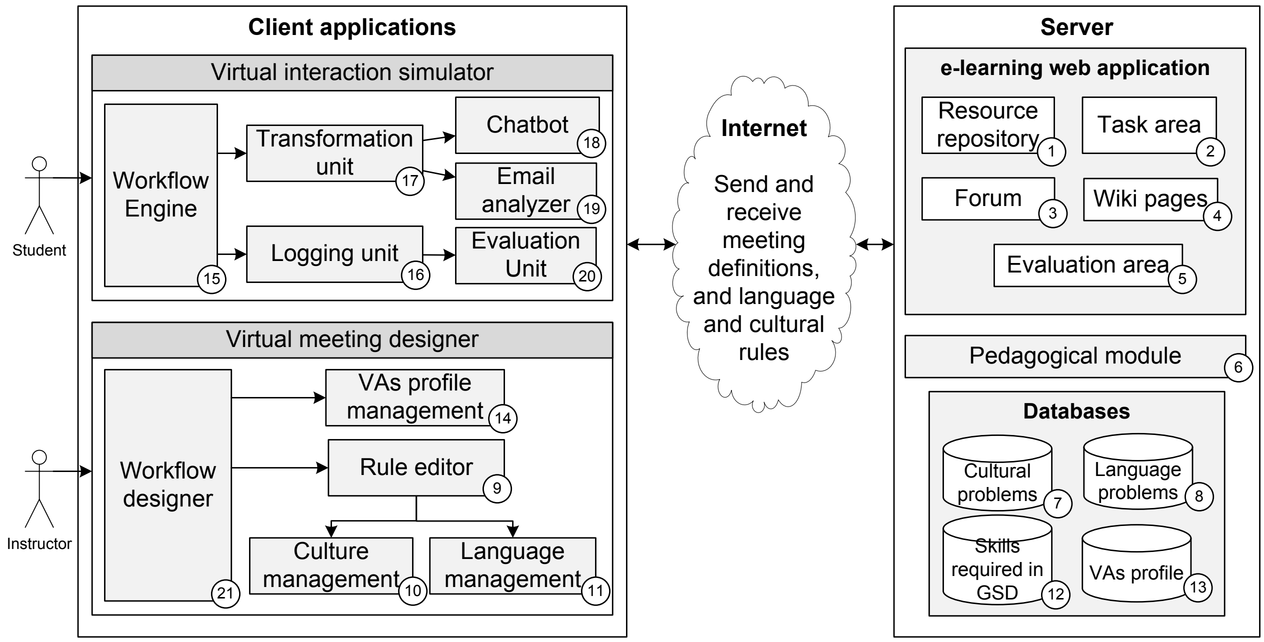

Figure 1. Architecture of VENTURE

B. *Cultural problems*

The *cultural problems* database (7) consists of a repository of textual structures that can be used in the simulators and which train cultural problems and recognized differences that might affect communication in GSD scenarios. These structures are formed in VTRML language and are stored with regard to the cultural dimension that they deal with so that they can be reused in the definition of new simulators. They also include a description of the difficulty, the seriousness of making each mistake and the actions that the VC will take to correct the student.

In order to prepare these scenarios, VENTURE has been designed to consider those cultural problems that are present in GSD such as: the Deaf Effect [26], the Mum Effect [27], the use of direct or indirect styles, etc. These problems can appear to higher or lower degrees depending on the cultural dimensions of the cultures which are present during an interaction. In this respect, we concentrate our efforts on the cultural dimensions presented by Hofstede [28], along with those of House [29] which are contained in a more recent model. The latter has defined various cultural dimensions according to the current characteristics of globalization, which we have adopted for our model. This model specifically consists of eight dimensions [29]:

- **Uncertainty Avoidance**: Degree to which the individual feels 'comfortable' in new situations. Members tend to avoid uncertainty by relying on social norms, customs, and bureaucratic practices to deal with unpredictable events. A common training simulator would consist of involving a student with a culture that presents a low degree of uncertainty avoidance in an interaction with VAs that show a strong resistance to change and to accepting their ideas. The student should, therefore find a way in which to politely persuade them.
- **Institutional Collectivism and Ingroup Collectivism**: Degree to which a community encourages and rewards the collective distribution of resources and collective action. This includes factors such as loyalty and cohesiveness of the individuals. For example, a student use a simulator to train in how to interact with VAs with low institutional collectivism that tend to be independent and not used to working in groups.
- **Assertiveness**: Degree to which individuals are dominant, confrontational, and aggressive in social relationships. A student from a highly assertive culture would need to learn to cooperate and have warm relationships through a simulator that would allow him/her to interact with VAs of low assertiveness.
- **Future Orientation**: Measures time aspects for the satisfaction of needs, including how the students will engage in future-oriented behaviours such as planning, delaying gratification and investing in the future. For example, a student with high future orientation interacting with low future individuals should learn how to provide gratification as soon as possible.
- **Human Orientation**: Degree to which individuals encourage and reward individuals for being fair, altruistic, friendly, tolerant, generous and supportive of each other. A student with low human orientation would need to train with a simulator in which s/he would interact with VAs in order to learn to show interest and affection.

- **Performance Orientation**: Degree in which members are encouraged in their tasks and rewarded for their achievements. Low performance orientation students prefer to use a more indirect style of speech while high performance members tend to be more competitive and direct. The instructor could, for example, decide to train a student with a direct communication style to communicate with low performance members in a cordial manner in order to improve team trust and motivation.
- **Power Distance**: Degree to which less powerful individuals agree that power should be shared unequally. A training activity for this dimension would be focused on establishing good relationships between participants in which authority is understood in different ways. For example, a student from a culture with a low power distance might find it difficult to accept authority when interacting with colleagues from high power distance cultures, in which classes are well differentiated. Or vice versa; a student from a high power distance culture would need to learn to interact in a less hierarchal manner.
- **Gender Egalitarianism**: This is the degree of gender equality. Simulations could teach a woman how to interact with cultures in which there is a great separation between male and female roles, while men from these cultures would learn how to interact with women with regard to their position and culture.

The study of these problems and how to train them through appropriate simulators not only requires the integration of the Hofstede and House core concepts, but also necessitates the study of methods with which to mitigate cross-cultural differences [30], along with collecting the experiences of actual members.

Figure 2 shows a fragment of the definition of the meeting in which the Spanish student and the Russian VA are interacting. According to the House dimensions, the Spaniards have a higher assertiveness and uncertainty avoidance, whilst Russians have a higher institutional collectivism and gender egalitarianism. In this example, we show two cultural structures for training highly assertive students to interact with low assertive individuals.

C. Language problems

The *language problems* database (8) contains the linguistic problems that may appear when participants interact textually with a non-native language.

As in the previous case, the structures that serve to train these problems are stored in VTRML structures that can be used in any simulator through their inclusion in the simulator definition by means of a wizard.

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, considering any kind of common linguistic problem among the languages involved. Some of the typical written linguistic problems when non-native speakers interact using English are[31].

- the incorrect use of "false friends" (when a word looks or sounds similar in two different languages, but differs in meaning)
- the overuse of certain verbs of high semantic generality (e.g. do, have, make, put, take)
- the incorrect formation of conditional clauses
- the incorrect use of verbal tenses
- the avoidance of passive forms
- the absence of the third person –s

An example of how we deal with linguistic problems can be seen in Figure 2. 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).

D. Skills required in GSD

The *skills required in GSD* are stored in the database (12) which contains best practices in the form of VTRML structures, as in the case of the cultural and language problems. This knowledge is classified into the following skills that they are intended to train and which, according to our systematic literature review, are needed in GSD:

- 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.

E. VAs profile database

The *VA profile (13)* contains the information regarding the virtual characters involved in the training scenarios (VAs and VCs), and defines their appearance, emotions and gestures. This is used to teach students how they should understand and react to different customs during a conversation, as depicted in Figure 3.

The *VA profile management* module (14) permits these profiles to be maintained so that new characters can be included or existing ones can be modified.

F. Workflow Engine

On the client side of VENTURE, the *Workflow Engine* (15) is responsible for executing the meeting workflows by interpreting the definition of the meeting, and orchestrating the corresponding phases.


```
<Workflow Name="Requirements Elicitation Meeting" duration="25:00">
  <Sequence>
    <Phase Name="Introduction">
        <!--Not detailed in this example"-->
    </Phase>
    <Phase Name="Client request">
        <ConvesationalKnowledge>
          <category order="1" gesture="concerned"><!--VC1 starts the conversation (order=1)-->
               <template VA="VirtualCustomer1"> We would need to have our components working with the
                           exporting functionality by next week</template>
               <template VC="VirtualColleague1"> We do not have experience in using these components.
                              You should answer negatively </template>
          </category>
          <category gesture="thinking">
              <pattern>*information * components * time</pattern> <!--The student must mention the
               need for information about the components and also ask for more time-->
            <action>NextPhase</action> <!—Reaching the previous pattern the next phase continues-->
          </category>
        </ConvesationalKnowledge>
        <CulturalProblems>
          <CulturalProblem type="Assertiveness" severity="3">
            <category gesture="thinking">
              <pattern>DEFAULT</pattern><!—Pattern triggered if the previous one doesn’t match-->
              <template VA=" VirtualColleague1">You should explain the reasons better and be more
indirect</template>
            </category>
          </CulturalProblem>
        </CulturalProblems>
    </Phase>
    <Phase Name="Data layer document">
      <ConvesationalKnowledge>
        <category gesture="concerned"><!--VirtualCustomer1 starts the conversation-->
           <template VA="VirtualCustomer1"> Ok, I will send you more information. Yesterday I did not
              receive your document with the proposal for the data layer. Did you have any
              problems?</template>
           <template VC="VirtualColleague1"> You should mention that we had technical
problems</template>
        </category>
        <category gesture="angry">
          <pattern>* problems * after|later|tomorrow|next|soon|days</pattern> <!--The student
explain the problems that s/he had and when is s/he going to send the deliverable"-->
          <action>NextPhase</action> <!—Reaching the previous pattern the next phase continues-->
        </category>
     </ConvesationalKnowledge>
        <LanguageProblems>
          <LanguageProblem type="false friend" severity="2">
            <pattern>red</pattern> <!--Incorrect use of the word "red"-->
            <template>“red” is a false friend in Spanish. It can be translated as “network” in
English. Is that what you mean?</template>
            </LanguageProblem>
        </LanguageProblems>
        <CulturalProblems>
          <CulturalProblem type="Assertiveness" severity="3">
            <category gesture="thinking">
              <pattern>* problems *</pattern> <!--The student doesn’t give detailed information"-->
                <template VA="VirtualColleague1"> You should tell him when you are going to send him
                                     the report </template>
            </category>
          </CulturalProblem>
        </CulturalProblems>
    </Phase>
  </Sequence>
</Workflow>
```

Figure 2. Definition of a simulator in VTRML language

The VTRML format contains the AIML language embedded in their definition. Figure 2 shows an example of the definition of a workflow. The *Workflow Engine* interprets this content and extracts the conversational knowledge, together with the linguistic and cultural rules defined.

The process of extracting this information in order to generate the AIML language is carried out by the *transformation unit* (17), so that the information obtained will be understandable by the *chatbot system* (18), in the case of synchronous interactions, and by the *Email analyzer* (19), in the case of dealing with asynchronous interactions.

In this respect, the main components are: the *chatbot system*, which processes the natural language by interpreting AIML, and the *workflow orchestrator*, which guides the execution of the workflow. The *login unit* (16) makes it possible to save the log of the conversation so that the instructor can review it later.

G. *Evaluation unit*

The main goal of the *Evaluation unit* (20) is to gather information about the course and the use of the simulators in order to provide an evaluation that would serve to determine what skills a student must improve in order to assign them the most appropriate training programs that will focus on their particular needs.

This module measures, in real time, factors such as: time taken in each theoretical module, evaluation of the practical exercises, delay in the upload of deliverables, etc., and in the case of the simulators it measures factors such as: average response time or number of corrections made by the VC in the case of the simulators.

H. *Workflow Designer*

Since the VTRM format may require in-depth knowledge for its edition, the *Workflow Designer* (21) allows the virtual meetings to be defined and modified in a graphical manner. The virtual meetings are thus designed as sequential workflows made up of a set of phases containing the specifics details of the conversation for that phase. Based on the graphical definitions of the meetings, this component automatically translates the definition of the workflow into VTRML format.

IV. VIRTUAL INTERACTIONS

The simulations can be carried out by means of the chat plugin which trains students in synchronous communication, or through the e-mail plugin which provides instruction in asynchronous communication.

A. *Synchronous Interactions*

A real synchronous interaction is displayed in Figure 3, in which the Spanish student is chatting with a virtual client from Russia. These simulations are time-limited, and the student can pass to the next phase of the conversation without completing it if s/he considers it appropriate to do so, or does not know how to deal with that situation. However, if the student makes an appropriate interaction, the transition to the following phase is automatic. The students' scope is to obtain as much information as possible within the time limit while attempting to deal with all the phases in order to successfully achieve their task by minimizing the cultural and language errors made.

B. *Asynchronous Interactions*

In the case of the asynchronous interactions, the workflows define the sequence of emails that will be exchanged during the simulation. In this case, the structure of the email is defined as an XML in which the different patterns are embedded using the AIML format. The analysis of the emails sent by the student consists of verifying that they are well organized and that they deal with the problem in the correct manner without committing cultural and language mistakes. In this case, the VC will correct the student by sending him/her an email reporting the mistakes made.

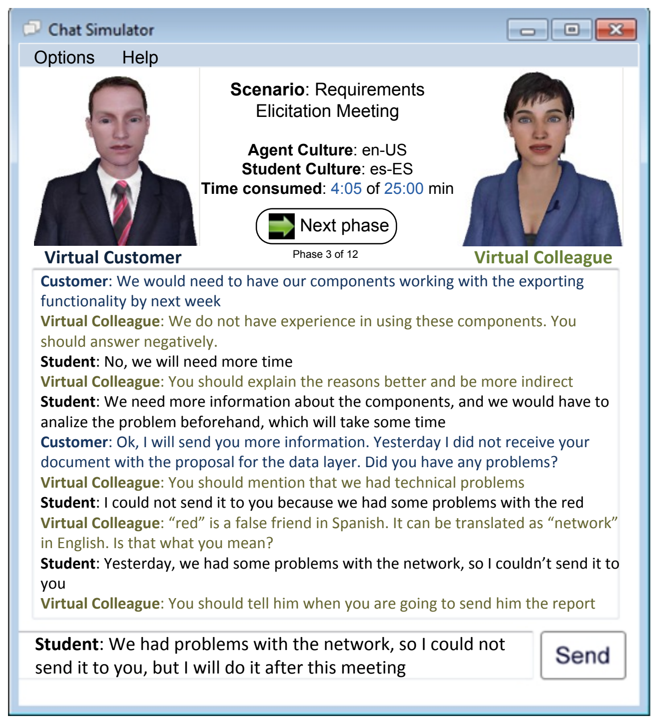

Figure 3. Example of the chat plugin interface

V. CONCLUSIONS

This paper presents a framework which is focused on providing training in GSD activities. The principal component of this framework is VENTURE, whose main objective is to deal with the skills required in GSD, specifically those related to cultural and linguistic diversity.

The chief advantage of using VENTURE is that students are more independent and they can train at any moment without depending on the availability of other partners or colleagues. VENTURE also provides the following advantages:

- Controlled training: The simulations are rigorously defined, and since the VAs control the conversations, the off-topic conversations which quite often take place when two or more students interact will never occur in VENTURE, since the VC will detect wrong interactions not related to what the student is expected to say.
- The reflection and reception of feedback is promoted not only by means of the corrections made by the VC, but also by any assessments that the instructors wish to make, since the tracking information of the student's progress is made available to them.
- It provides repeatable training, since the students can practice the scenario in the same conditions and confront the same problems when they consider that they need more practice in order to learn something.
- Safety of the simulation: No real projects run the risk of involving inexperienced participants.
- Situations which are difficult to reproduce or which could create conflicts in real interactions between students can be practiced.
- Practice from the perspective of different roles, in order to learn about the different kinds of problems that may occur from different perspectives.
- Reduction of the instructors' effort involved in coordinating practices with distant institutions.

VI. RESEARCH AGENDA

We plan to collect evidence, not only from various universities but also from professional environments, that will allow the effectiveness of the framework to be demonstrated.

In this stage, the VENTURE's *evaluation module* will be employed to measure its use, since it automatically tracks several aspects of the students' activities, providing a mechanism with which to systematically evaluate the framework. We are also in the process of preparing surveys, structured interviews and in-situ observations in order to obtain feedback with which to improve the framework.

Issues such as ensuring students' engagement may be barriers to the eventual effectiveness of the application, and we will therefore focus these evaluations in order to guide our future evaluation efforts towards answering the following research questions:

- Do the students understand the purpose of the simulations?
- Do the students feel that they have improved their skills in GSD? Do they develop their communicative abilities? Do they acquire new syntax and vocabulary related to the topics?
- Do the students achieve the learning objectives at the end of the course? How long do the students need to complete the course and to deliver the artifacts?
- Do students feel motivated when interacting with VAs? To what extent does a student perceive the usefulness of the framework?
- What is the students' opinion of the usability of the tool?
- Do students identify with their roles in the virtual simulations? What problems occur during the interactions with VAs?

We also wish to compare the performance of students who have trained with our framework with that of others of the same characteristics who have not. In order to measure this, it would be useful to study communication metrics for software development [32], that could help to measure and compare the adequacy of the communications in both cases.

From the point of view of the instructors, we must also study the feasibility of the framework for designing and customizing the training materials and simulators. In this respect, their opinions will be useful to discover the answers to the following questions:

- What is the instructors' opinion of the usability of the simulators' designer? How long does it take them to design a simulator?
- What is their perception of the monitoring and evaluation facilities? Do they gather useful information?
- Were the rules and knowledge stored in VENTURE reusable and transferable to other courses?

The results of these evaluations will eventually help us to design better simulators and to improve the cultural and linguistic rule databases, along with the set of structures used to train the GSD skills. Finally, we also intend to test the usability of the architecture by asking two final year students to develop simulators by means of the designer. We will thus attempt to demonstrate that VENTURE is a generic architecture that can be used in an intuitive mode and can be adapted to different learning scopes, principally those which are distributed.

ACKNOWLEDGMENTS

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 the Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), and co-funded by Fondos FEDER, in addition to MELISA (PAC08-0142-3315), the Junta de Comunidades de Castilla-La Mancha, the Consejería de Educación y Ciencia in Spain, and ORIGIN (IDI-2010043 (1-5)) funded by CDTI and FEDER.

REFERENCES

[1] M. J. Monasor, A. Vizcaíno, and M. Piattini, "Preparing students and engineers for Global Software Development: A Systematic Review," In: *International Conference on Global Software Development (ICGSE 2010)*, Princeton, NJ, USA, 2010, pp. 177-186.

[2] F. Ciccozzi , and I. Crnković, "Performing a project in a Distributed Software Development Course: Lessons Learned," In: *International*

Conference on Global Software Development (ICGSE 2010), Princeton, NJ, USA, 2010, pp. 187-191.

[3] M. Fitze, "Discourse and Participation in ESL Face-to-Face and Written Electronic Conferences," *Language Learning & Technology*, vol. 10, no. 1, pp. 67-86 2006.

[4] E. Bradner , and G. Mark, *Why distance matters: effects on cooperation, persuasion and deception*. New Orleans, Louisiana, USA: ACM, 2002.

[5] A. Hauer , and M. Daniels, "A learning theory perspective on running open ended group projects (OEGPs)," In: *Conferences in Research and Practice in Information Technology*, vol. 78, Simon and M. Hamilton, Eds. Wollongong, NSW, Australia: ACS, 2008, pp. 85-92.

[6] P. Lago, H. Muccini, L. Beus-Dukic, I. Crnkovic, S. Punnekkat, and H. V. Vliet, *Towards a European Master Programme on Global Software Engineering*: IEEE Computer Society, 2007.

[7] M. Daniels, "The contribution of open ended group projects to international student collaborations," *ACM Inroads*, vol. 1, no. 3, pp. 79-84 2010.

[8] M. Daniels, 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* vol. 26, no. 4, pp. 795-806 2010.

[9] E. Keenan, A. Steele, and X. Jia, "Simulating Global Software Development in a Course Environment," 2010, pp. 201-205.

[10] A. Berglund, *Learning computer systems in a distributed project course: The what, why, how and where*. Uppsala: Uppsala University, 2005.

[11] C. Laxer, M. Daniels, Å. Cajander, and M. Wollowski, *Evolution of an international collaborative student project*. Wellington, New Zealand: Australian Computer Society, Inc., 2009.

[12] R. Prikladnicki , and L. Pilatti, *Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience*. Bangalore, India: IEEE Computer Society, 2008.

[13] M. Hackett, *Building Effective Global Software Test Teams through Training*. Munich, Germany: IEEE Computer Society, 2007.

[14] B. Lutz, *Training for Global Software Development in an International "Learning Network"*: IEEE Computer Society, 2007.

[15] A. Braun, A. H. Dutoit, A. G. Harrer, and B. Brüge, *iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses*: IEEE Computer Society, 2002.

[16] R. Ocker, M. B. Rosson, D. Kracaw, and S. R. Hiltz, "Training Students to Work Effectively in Partially Distributed Teams," *Trans. Comput. Educ.*, vol. 9, no. 1, pp. 1-24 2009.

[17] Å. Cajander, T. Clear, M. Daniels, J. Edlund, P. Hamrin, C. Laxer, and M. Persson, "Students analyzing their collaboration in an International Open Ended Group Project," In: *39th ASEE/IEEE Frontiers in Education Conference.*, San Antonio, Texas 2009, pp. T1A1-T1A6.

[18] D. Corder , and A. U, "Integrating Second Life to enhance global intercultural collaboration projects," *ACM Inroads*, vol. 1, no. 3, pp. 43-50 2010.

[19] T. Clear , and M. Daniels, "2D & 3D Introductory Processes in Virtual Groups," In: *33rd ASEE/IEEE Frontiers in Education Conference.*, Boulder, Colorado, 2003, pp. S1F1-SIF6.

[20] B. S. Hasler, T. Buecheler, and R. Pfeifer, *Collaborative Work in 3D Virtual Environments: A Research Agenda and Operational Framework*. San Diego, CA: Springer-Verlag, 2009.

[21] T. Schümmer, S. Lukosch, and J. M. Haake, *Teaching distributed software development with the project method*. Taipei, Taiwan: International Society of the Learning Sciences, 2005.

[22] A. Soller, "Supporting Social Interaction in an Intelligent Collaborative Learning System," *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 12, pp. 40-62 2001.

[23] T. Clear , 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, Eds. Portland, Oregon: ACM, 2008, pp. 63-67.

[24] M. Daniels, M. Petre, V. Almstrum, L. Asplund, C. Bjorkmann, C. Erickson, B. Klein, M. Last, and A. Berglund, "RUNESTONE, an International Student Collaboration Project," In: *IEEE Frontiers in Education Conference*, Tempe, Arizona, 1998.

[25] D. E. Leidner , and S. L. Jarvenpaa, "The use of information technology to enhance management school education: a theoretical view," *MIS Q.*, vol. 19, no. 3, pp. 265-291 1995.

[26] D. Robey, "Blowing the whistle on troubled software projects," *Commun. ACM*, vol. 44, no. 4, pp. 87-93 2001.

[27] S. Ramingwong , and A. S. M. Sajeev, *A Multidimensional Model for Mum Effect in Offshore Outsourcing*: IEEE Computer Society, 2008.

[28] G. Hofstede , and G. J. Hofstede, *Cultures and organizations: software of the mind*, New York, NY, USA, 2005.

[29] R. J. House, P. J. Hanges, M. Javidan, P. Dorfman, and V. Gupta, *Culture, Leadership, and Organizations: The GLOBE Study of 62 Societies*: Sage Publications, 2004.

[30] K. Jablokow , and M. Myers, *Managing Cognitive and Cultural Diversity in Global IT Teams*. Princeton, NJ, USA, 2010.

[31] P. Shachaf, "Cultural diversity and information and communication technology impacts on global virtual teams: An exploratory study," *Inf. Manage.*, vol. 45, no. 2, pp. 131-142 2008.

[32] B. Bruegge , and A. H. Dutoit, *Communication metrics for software development*. Boston, Massachusetts, United States: ACM, 1997.