Providing Training in GSD by Using a Virtual Environment

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

Abstract. In Global Software Development (GSD) the human factor is one of the main assets for the companies. Their efficiency in communication and collaboration, as well as their knowledge of the processes applied in GSD, can lead the companies to be more competitive.
Participants require knowing the customs and culture of other participants. Moreover they need to improve their social and interpersonal competencies such as: negotiation, teamwork, conflict resolution, time management, leadership, and communication skills using a common language.
In this paper we present a simulation-based approach for training GSD with which users can train by interacting with Virtual Agents which play a role in the development process. These Virtual Agents textually interact with users by means of a chat by simulating being people with different personalities, experiences, skills and culture.
The lessons learned in a feasibility study carried out with a group of practitioners and PhD students are also analyzed in this paper.

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

1 Introduction

In Global Software Development (GSD) the human factor is one of the main assets for the organizations [1]. Their productivity, the effective use of communication and collaboration tools and their knowledge of the processes can lead the companies to be more efficient than their competitors [2]. Because face-to-face contact is restricted in GSD, human relations and attitudes towards distant participants become a key factor.

Therefore, participants must not only focus on technical aspects, but also in social and interpersonal and intercultural competencies such as: negotiation skills, teamwork skills, conflict resolution, time management skills, leadership, decisions making, reasoning skills, knowledge of communication protocols and customs, communication skills using a common terminology in a second language, ability to motivate others and create trust, information management, ambiguity and uncertainty management and knowledge of tools and processes used in GSD [3, 4].

Personality dimensions such as extroversion, anxiety, self-control, sensitivity, independence, emotional stability, reasoning or dominance also influence in software activities success [2, 5].

In terms of cultural boundaries, there is also a need for cultural understanding and sensitivity. Participants require knowing the customs and culture of other project members, as this factor significantly influence the performance and quality of their work [6]. For instance, in some cultures participants tend to not to speak during team meetings until invited to do so, because it would be considered impolite in their culture. Therefore, a software engineer should be trained to detect what problems can arise in this type of development and for instance to know how to confront situations to kindly encourage such participants to discuss a certain issue.

On the other hand, English is used more and more in international collaboration, gaining importance in applied linguistics research given the challenges for non-native speakers. Lack of fluency is a common problem which produces hesitancy or delay [7]. Native speakers have a natural tendency to assume facts that can negatively affect the project and eventually damage the team relationships. However, non-native speakers must improve their language proficiency and learn common expressions that could allow them to participate in a more dynamic way.

Our goal is to help software engineers (practitioners or students) to develop all these skills necessary for GSD. Thus, first of all we carried out a systematic literature review [4] were we found that the main difficulty for providing an adequate training on these skills consists of setting up realistic settings that could allow the students to tackle representative problems. Therefore, the current educational programs rarely organize collaborative activities with other institutions, because of scheduling differences and coordination problems [8].

To tackle that problem we propose a simulation-based approach for training some of the problems that often take place in GSD. Therefore, users can train at any moment by interacting with Virtual Agents which play a role in the software development process. These Virtual Agents interact textually with user by means of a chat by simulating being people with different personalities, experiences, skills and culture. In this way, it is possible to design rigorous training scenarios for dealing with specific problems and skills required in GSD.

The architecture of the simulator is integrated into an e-learning platform and it has also been designed to ease the design of the simulation scenarios. The definition of the simulation scenarios is based on a metadata language defined for this purpose. The scenarios designer permits to drag and drop the different phases in which the simulation is composed by defining a sequential workflow that will guide the conversation. Each phase contains the conversational knowledge required for such specific context as well as cultural and linguistic rules that will allow correcting the users' interactions by means of a special virtual agent called virtual colleague that will help the users during the simulation.

The execution engine is responsible for interpreting the defined information and executing the different phases of the conversation within the time limit. For that purpose it uses a chatbot system that responds to the users. The students' goal is to confront situations in which they could appreciate typical problems, detecting cultural differences and trying to obtain as much information as possible during the conversation, as well as minimizing the cultural and language errors made during the interaction.

Other advantages of the proposal are that it provides an independent and controlled training, the rapid reception of feedback by means of the virtual colleague, the rigor of the training in cultural issues and the reduction of the instructors' effort.

A feasibility study was carried out in a presentation of the tool that served to analyze the first impressions of experts of multinational companies as well as of a set of PhD students that provided feedback to improve several aspects of the architecture and the training scenarios.

This paper is organized as follows: Section 2 explains the context of this research. The influence of personal aspects in GSD is explained in Section 3. The proposal in which is focused this work is detailed in Section 4. Section 5 provides an example of its use. Finally, Section 6 provides some concluding remarks and future work.

2 Context of current research

In the context of our research, the main focus is around the influence of social factors and cultural and linguistic differences in GSD. According to a systematic literature review on the field of teaching and training GSD [4], the strategies that have been applied in academic courses, mainly consists of replicating the conditions of real environments by collaborating in software developments with other universities [9]. GloSE-Lab [8] is an example including theory and practice in collaboration with distant universities.

The main problems of these approaches are given by the difficulty of creating settings that suits the teaching goals of each university, which moreover, can have different schedules. The lack of knowledge and experience in the development of big projects, the unequal workload of the participants, communication problems and conflicts among partners are common problems.

Blended learning environments [10] and e-Learning platforms [11] are commonly used in these approaches in order to facilitate the collaboration among students from distant universities, while other approaches are focused on games and simulation, placing the users in scenarios in which they have to cope with specific problems of GSD. As the context of our research is in this last field, the following subsection details the related state-of-the-art.

A. Games and simulation

Learning happens most easily when the students actually need the knowledge of how to do something for a reason. Feedback reception is an important factor to enhance knowledge and skills of the student [12]. Games and simulation have been applied in many fields of software engineering because these approaches are among the most motivating for the students.

An example of on game-based approaches is IT Billionaire [13]; a turn based board game designed to teach dynamics of GSD in order to discover the many variables involved in these environments. The players must attempt to become billionaire by running a company that applies GSD. However these kinds of approaches are limited to develop a reduced set of skills mainly related to management tasks.

In terms of simulation, the SESAM project [14] is a representative example intended for investigating and comparing different strategies for software development. Students use a textual interface in which they read and type text for training in project management activities.

M. Samejima et. al [15] address situation-dependent scenarios, in this case for simulating project management activities, specifically covering generation of scenarios for the progress management phase.

The use of augmented spaces like iBistro [10], based on the 'learning by doing' approach are also used as a way to enable distributed members to collaborate during the development so that can be used to learn project management, software development and social skills.

Social aspects are a key factor for improving the development process [2], and in this field, M. Yilmaz and R. V. O'Connor [16] propose a framework for modeling development activities, serving for the research of several social issues in software development, such as team formations, interpersonal conflicts or social loafing.

In a related vein, C. Pelachaud [17] has worked on behavior expressivity, presenting an affective embodied conversational agent which is able to display communicative and emotional signals. Expressive qualifiers are used to modulate the expressivity of lip movement [18].

An embodied conversational agent is presented by M. Kavakli et. al [19] with the aim of counseling neglected aborigines in Australia, who have problems of poverty and disease as a consequence of past neglect and torture. The agent plays the role of sociologist in advising on strategies to overcome their addiction to alcohol problems. It can also represent protocols to express social relations between humans.

In the field of GSD, [20] explores the interaction with avatar-based humans in virtual collaborative projects, in order to train collaboration skills and intercultural competences. Configurable avatars are also used in Teamlink [21]; a collaborative 3D virtual environment conceived to support icebreaking activities with the aim of establishing trust between virtual team members.

B. Teaching cultural aspects

GSD is recognized as a sociotechnical activity in which cultural play an important role. Practitioners cannot ignore the impact of cultural diversity and the barriers and problems it can create [22]. Educators must prepare the students, at undergraduate and postgraduate level to tackle the problems in these environments, paying special attention to the impact in computer mediated communication, which is particularly affected by this aspect [23].

Many organizations try to minimize this impact by implementing strategies in order to foster an organizational culture trying to set up a set of norms, values, objectives and beliefs that are touch to their members [23]. However, many aspects of the national culture are difficult to change, and participants must learn to understand each other character, philosophy and mindset [24]. The following authors propose different ways in which to classify cultural aspects:

- **Hofstede** [25]: defines a classification focused on the values and culture of computer professionals, considering five value dimensions in which countries differ: power distance, uncertainty avoidance individualism/collectivism, masculinity/femininity and long-term/short-term orientation.
- **House et al.** [26]: is a more recent classification focusing on culture and leadership in 61 nations, defining the following dimension: uncertainty avoidance, power distance, societal collectivism, in-group collectivism, gender egalitarianism, assertiveness, future orientation, performance orientation and humane orientation. The first six dimensions have their origins in the Hofstede dimensions.
- **Trompenaars and Hampden-Turner** [27]: compare culture to an onion made up of layers that can be peeled to be understood. They outline seven dimensions of culture: universalism versus particularism, individualism versus communitarianism, "specific" versus diffuse, affective versus neutral, achievement versus ascription, sequential versus synchronic, internal versus external control.
- **Hall** [28]: for whom culture is equated to communication, which is made up of three elements: words, material things and behavior. He defined seven relevant concepts to study national and corporate culture: speed of messages, context, space, time, information flow, action chains and interfacing.

By considering these dimensions it is possible to quantify the probability that certain cultural patterns could occur during the interaction, and this may serve to focus the training for confronting specific patterns. As an example, E. MacGregor et. al [29] present a set of cultural patterns in GSD: yes (but no) pattern, proxy pattern, we'll-take-you-literally (anti) pattern, we're-one-single-team (anti) pattern, the-customer-is-king (anti) pattern.

For example the "Yes (but no) pattern" consists in the fact that individuals in some cultures tend to respond requests by saying "yes". As a consequence, problems may appear if the person who made the request trusts in that false answer. In order to minimize the problems that these kinds of patterns can cause, software engineers must receive a rigorous training by considering their culture.

3 Influence of personal aspects and skills in GSD

Personal aspects play an important role in GSD, due to the fact that people make decisions by mixing feelings and logic in a different degree depending on their values, knowledge and personality [30]. As an example, extrovert people tend to pay attention on the external environment, while introverts are more focused on the internal environment when taking decisions [31].

Some studies have found that virtual team performance was directly related to leadership and interpersonal dimensions, which are influenced by personality and psychology factors [32]. Moreover, factors such as personality or charisma of an individual can also affect the overall team performance [33].

These factors are closely influenced by the participants culture, as well as their age, gender, experience and region, and the team size [33]. Moreover these attitudes directly affect colleagues' satisfaction, effectiveness and performance, all the more when the practitioner is in a position that requires leadership abilities. Extroverted behaviors, for example, tend to result in higher frequency of communication through electronic messages and an increase in the team performance [34]. In [35] it was found that high levels of positive personal traits, such as helpfulness and agreeableness increased team performance satisfaction.

GSD involves high requirements for communication and collaboration between its practitioners in a common language. Fluency is also a problem which produces misinterpretations, hesitancy or delay [22]. Native speakers have a natural tendency to assume facts that can affect negatively in project. These attitudes can lead to the loss of pertinent information and eventually damaging the team relationships being necessary to train the skills that allow to minimize these problems and also to deal with them when they occur.

Instant messaging, mail, phone, and video conferencing systems are the main media for both formal and informal communication in GSD [36], so it is necessary to have additional skills in their use in order to build trust and social relationships with co-workers by considering the characteristics of each communication mean. The aim of this proposal consists on providing training in instant messaging and mail communication considering some of the aforementioned personal factors.

4 VENTURE

The aim of this research work is to define a tool for providing training in same of the skills required in GSD, considering cultural and personal factors that can influence the development process. VENTURE (*Virtual ENvironment for Training cUlture and language problems in global softwaRe dEvelopment*) is intended to cover some of the following aspects in the training of the skills required in GSD:

The chat simulator would make it possible to create simulations with which to achieve the following:

1. Creating awareness of the different kinds of cultures and the problems that may appear during textual interactions.
   - Showing the typical language mistakes of the people from a particular culture.
   - Showing the different gestures, customs and behaviors of people from other cultures that could be misunderstood.
   - Showing ways in which to ask questions in an appropriate manner.
   - Using direct/indirect communicative style.
   - Using formal and informal communication skills.
2. Promoting skills for the development of relationships based on cultural diversity
   - Identifying issues that may cause conflict.
   - Showing tips to assist in reacting towards certain actions or to answer specific questions.

    - Teaching strategies by which to eliminate bias and discrimination. For instance avoiding the usage of sexist words.
    - Showing ways in which to establish a dynamic conversation.
3. Training the proper use of a second language
    - Ability to effectively communicate with multidisciplinary members using the same terminology.
    - Minimizing language mistakes and teaching structures to allow people to communicate in an effective manner.
4. Developing teamwork skills
    - Leadership and decisions making skills.
    - Time management skills.
    - Showing how to develop mutual trust and confidence.
    - Knowledge of negotiation skills.
    - Management of ambiguity and uncertainty.
    - Conflict resolution skills and critical reasoning skills.

VENTURE provides a platform in which users will acquire practice in these aspects by means of simulation. Users will confront common situations in which they could appreciate the problems of interacting in the distance with people with different personalities, experiences, skills and culture. For achieving this it uses Virtual Agents (VAs) playing a specific role in the Software Engineering process, as a mean to simulate any kind GSD scenario. These VAs are characterized by a specific culture and personality, and will textually interact with the student. Moreover, during the interactions, a Virtual Colleague (VC) will guide the user to address the simulation, giving advice and correcting the inappropriate interactions.

This section shows the main components of VENTURE (see Fig. 1), which follows a client-server architecture. An e-learning application is the core of the server side and 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.

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.

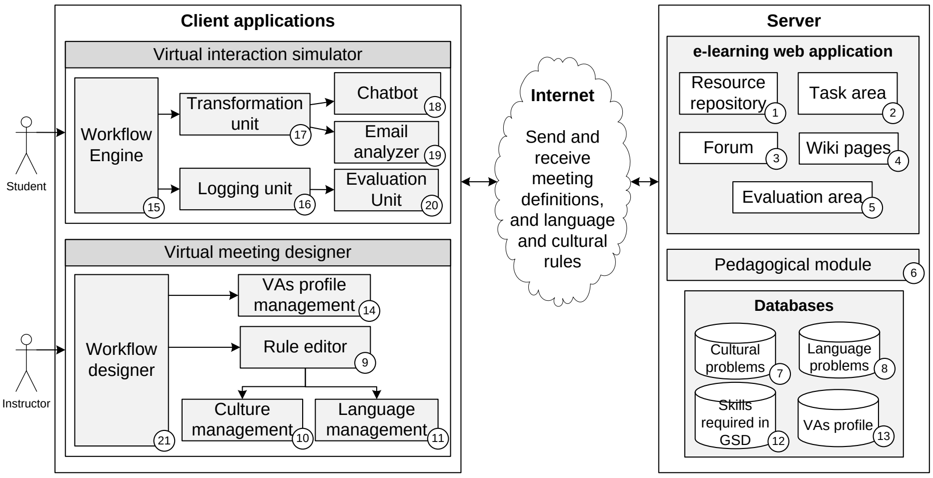


Figure 1: VENTURE Architecture.

Cultural problems

The *cultural problems* database (7) contains the cultural rules that can be used in the simulations and which train cultural problems that might affect communication in GSD scenarios. They consist of VTRML (VenTuRe Markup Language) structures; a language specifically defined for VENTURE, containing the rules for each phase of the simulation as well as the necessary metadata required for the execution engine for the simulation.

Language problems

The *language problems* database (8) contains the linguistic problems that may appear when participants interact textually using a non-native language. As in the previous case, the structures that serve to train these problems are defined in VTRML, and can be used in any simulator through their inclusion its definition by means of a wizard. The linguistic rules are classified by 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).

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.

VAs profile database

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

Workflow Engine

The *Workflow Engine* (15) is responsible for executing the scenarios by interpreting the definition of the workflows, and orchestrating the corresponding phases. This engine interprets the VTRML definition of the workflows by extracting the conversational knowledge, together with the linguistic and cultural rules defined. This process is carried out by the *transformation unit* (17).

The conversational knowledge embedded in the workflow is defined in AIML language [37], which is interpreted 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.

The *login unit* (16) makes it possible to save the log of the conversation so that the instructor can evaluate it later.

Evaluation unit

The *Evaluation unit* (20) gathers information about the course of the simulations in order to provide an evaluation that would serve to determine what skills a student must improve.

Workflow Designer

The *Workflow Designer* (21) allows the virtual meetings to be defined and customized 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 each phase. Based on these graphical definitions, the definition of the meetings is automatically translated into VTRML format.

5 Simulating an interaction

In this section a training scenario based on a real experience by members of a company is presented. In this training scenario, a Spanish user called Alberto, plays the role of developer and has to interact with another developer from Germany called Georg. Georg has developed a webservice, and Alberto has (supposedly) developed and application that have to consume that webservice. However, Alberto has problems to consume the service because it does not follow the WSDL standard. The task for the Alberto in this scenario consists on explaining Georg what the problem is and what changes would be necessary in the webservice for solving the problem.

The training is intended to train the following specific skills: questions formularization, negotiation skills, trust creation and linguistic problems in the context of the conversation. Before starting the simulation, the situation is explained to Alberto, so he has an idea of how he is going to interact in order to convince Georg, although the VC will guide him at any moment.

By attending to the cultural dimensions of House [26], for Germany and Spain, the differences in the cultural dimensions are depicted in Table 1 (on the 1-to-7 scale, where 1 is the lowest value of fulfillment for a dimension).

Table 1: Cultural dimensions for Germany and Spain. Summarized from [26].

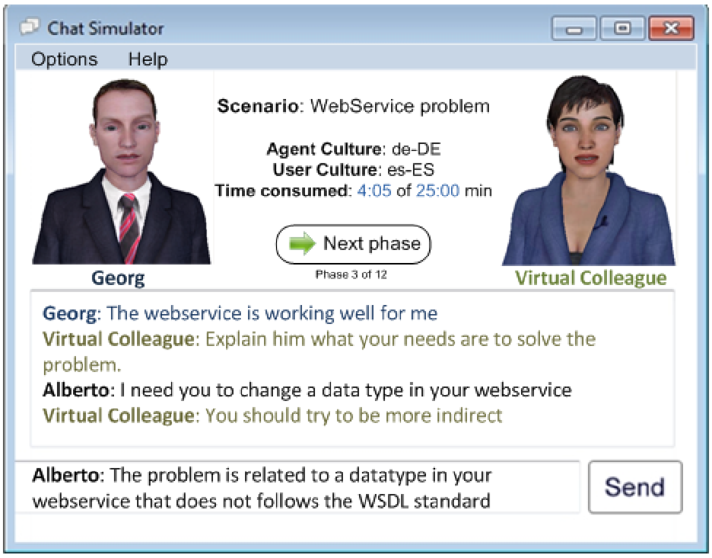
House Dimensions/County	**Germany**	**Spain**	**Difference**
Assertiveness	4.66	4.39	0.27
Institutional Collectivism	3.97	3.87	0.1
In-Group Collectivism	4.16	5.53	**-1.37**
Future Orientation	4.41	3.52	0.89
Gender Egalitarianism	3.25	3.06	0.19
Human Orientation	3.30	3.29	0.01
Performance Orientation	4.42	4.00	0.42
Power Distance	5.48	5.53	-0.05
Uncertainty Avoidance	5.35	3.95	**1.4**

Considering that the cultural dimensions that differ more between these two cultures are in-group collectivism and uncertainty avoidance, Alberto must interact in a proper manner to cover the problems that these differences can entail:

- **In-group Collectivism** is the degree to which a community encourages and rewards the collective distribution of resources and collective action, including factors such as loyalty and cohesiveness of the individuals [26]. Members of individualistic cultures tend to be direct in their communication, expressing their inner opinions, whereas collectivist cultures tend to be more indirect. In this simulation Alberto will try to interact in an indirect manner, and the scenario will be focused this. Fig. 2 show an example of a fragment of a conversation in which Alberto interacts in a too direct way and the VC corrects him.

In this case, the VC detects a direct intervention based on the detection of the patterns: “I need”, “you must”, “you have to”, etc. In case of detecting one of these patterns in the context of this part of the conversation, the VC will intervene to provide feedback. This rule is modeled in VENTURE as it follows, where the severity value is used for evaluation purposes, indicating the penalty that will receive the user if this rule is triggered:

```
<Skills type="direct-indirect style" severity="1">
    <pattern>"I  need  *"</pattern>  <!—Formulate  a  request  in  a
    direct style”-->
    <pattern>"You must *"</pattern>
    <pattern>"You have to *"</pattern>
    <template>You should try to be more indirect</template>
 </Skills>
```



Scenario: WebService problem

Agent Culture: de-DE
User Culture: es-ES
Time consumed: 4:05 of 25:00 min

Georg: The webservice is working well for me
Virtual Colleague: Explain him what your needs are to solve the problem.
Alberto: I need you to change a data type in your webservice
Virtual Colleague: You should try to be more indirect

Alberto: The problem is related to a datatype in your webservice that does not follows the WSDL standard

Figure 2: Fragment of a conversation in the chat interface.

- **Uncertainty Avoidance** is the degree to which the individuals feel 'comfortable' in new situations. Individuals tend to avoid uncertainty by relying on social norms, customs, and bureaucratic practices [26]. Individuals with high levels of uncertainty avoidance, tend to seek more feedback than those that are more tolerant to uncertainty, either by asking questions or observing. As Germans are less tolerant to ambiguity, Georg could feel anxiety and stress if Alberto is not clear with his proposal for a solution. Short-term feedback is a proactive method that Alberto should apply to avoid these feelings. Alberto should also try to minimize the uncertainty trying to give as much information as possible, avoiding, for example, misunderstandings or improper use of language. In the following fragment of a conversation, Alberto commits a mistake that could cause a misunderstanding:

Alberto: Could you realize some changes in the webservice?
Virtual Colleague: "Realize" is a false-friend in Spanish. Do you mean to "carry out"?
Alberto: Could you carry out some changes in the webservice?
Virtual Colleague: You should try to be more polite using "please"
Alberto: Please, could you carry out some changes in the webservice?

In this case, a language rule has been triggered in order to correct a false-friend mistake. Moreover, in the same conversation, the VC has also detected that the formulation of a question has been too direct without using the word "please". These rules are modeled as it follows in VENTURE:

```
<LanguageProblem type="false friend" severity="2">
   <pattern>realize</pattern> <!--Incorrect use of the word
   "realize"-->
   <template>“Realize” is a false-friend in Spanish. Do you mean to
   “carry out”?</template>
 </LanguageProblem>

<Skills type="politeness" severity="3">
   <pattern>"!please + ?"</pattern> <!—Formulate a question without
   “please”-->
   <template>You should try to be more polite using
   “please”</template>
 </Skills>
```

The VA (Georg) can also commit mistakes during his interaction, so that Alberto must receive a certain training of what he can confront in this sense. In the following example, the VC warns him about that mistake:

Georg: I have not become any request on my webservice.
Virtual Colleague: Note that he has incorrectly used the word “become”, when he wanted to say “received”.

In this case “become” is a false-friend, related to the german word “bekommen”, which means “to get or receive”. So Alberto must know how to manage the uncertainty that generates this kind of answer. The training scenario can be designed to detect the word “become” in this specific context of the conversation and teach the problem to the user. Moreover, the text introduced by the user is automatically checked by a spelling dictionary which will provide him feedback in case of committing grammatical errors.

6 Conclusions and future work

In this work we present a training environment for providing rigorous training in the skills required in GSD activities. A feasibility evaluation of VENTURE, was carried out by four experienced members of multinational companies, that provided insights for improving some aspects of the environment. After presenting them the environment, they were interviewed about their perception with regard to the use of the environment in their companies, its usefulness, and its usability from the students' and instructors' perspectives.

They all 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. Two responders stated that one of the main difficulties that companies usually encounter when organizing training courses is related to the difficulty in finding experts and the time needed to develop these courses. For these reasons they believed that the use of a simulator was a good idea, because these problems might be avoided. The flexibility that would make it possible to improve the training scenarios was also well valued.

The future work will be mainly focused on the evaluation of the environment. Our preliminary planning includes testing our approach on various experimental settings in which students, engineers and conference participants will be invited to participate in order to provide feedback about the perceived usability, motivation and adequacy of the scenarios. We are therefore preparing surveys and structured interviews with the following goals:

Validation goal 1: effectiveness and efficiency of VENTURE. The effectiveness and efficiency can be determined by analyzing the data gathered during the simulations, and the correction of deliverables after the simulation.

Validation goal 2: evaluation of the degree of adaptability of VENTURE to different cultures, languages and training goals.

Finally, we are also planning to carry out experiments with students at universities in Spain, Ireland, Mexico and Germany so that we could compare their performance with other students of the same characteristics that have not used this simulation tool. Therefore, several scenarios must be designed to train students of these different cultures considering specific goals in their training. For this purpose, we will also count with the experience of professionals that will guide us to provide realistic training scenarios based on real cases.

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 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 MJ, Piattini M, Vizcaíno A. Challenges and Improvements in Distributed Software Development: A Systematic Review. Advances in Software Engineering;2009:1-16 (2009)
2. Acuna ST, Juristo N, Moreno AM, Mon A. A Software Process Model Handbook for Incorporating People's Capabilities: Springer-Verlag New York, Inc., (2005)
3. Guzmán JG, Ramos JS, Seco AA, Esteban AS. How to get mature global virtual teams: a framework to improve team process management in distributed software teams. Software Quality Control;18(4):409-435 (2010)
4. Monasor MJ, Vizcaíno A, Piattini M, Caballero I. Preparing students and engineers for Global Software Development: A Systematic Review. International Conference on Global Software Development (ICGSE 2010); 2010 August 23-26, 2010; Princeton, NJ, USA: IEEE Computer Society. p. 177-186 (2010)
5. Acuna ST, Juristo N, Moreno AM. Emphasizing Human Capabilities in Software Development. IEEE Softw;23(2):94-101 (2006)

6. Abufardeh S, Magel K. The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges. 4th International conference on New Trends in Information Science and Service Science (NISS). Gyeongju, South Korea: 133 – 138 (2010)
7. Bordyuk L. Linguistic and culture-specific factors for professional success. CAD Systems in Microelectronics, 2003 CADSM 2003 Proceedings of the 7th International Conference The Experience of Designing and Application of; 2003 18-22 Feb. 2003. p. 530-532 (2003)
8. Deitersy C, Herrmannz C, Hildebrandtz R, Knauss E, Kuhrmannx M, Rauschy A, et al. GloSE-Lab: Teaching Global Software Engineering. International Conference on Global Software Engineering (ICGSE). Helsinki, Findland: 156-160 (2011)
9. Clear T. Replicating an 'Onshore' Capstone Computing Project in a 'Farshore' Setting – an Experience Report. International Conference on Global Software Engineering (ICGSE). Helsinki, Findland: 161-165 (2011)
10. Braun A, Dutoit AH, Harrer AG, Brüge B. iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses. Proceedings of the Ninth Asia-Pacific Software Engineering Conference. Gold Coast, Australia: IEEE Computer Society: 197 (2002)
11. Swigger K, Aplaslan FN, Lopez V, Brazile R, Dafoulas G, Serce FC. Structural factors that affect global software development learning team performance. Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research. Limerick, Ireland: ACM: 187-196 (2009)
12. Mandl-Striegnitz P. How to successfully use software project simulation for educating software project managers. Proceedings of the Frontiers in Education Conference, 2001 on 31st Annual - Volume 01: IEEE Computer Society: T2D-19-24vol.11 (2001)
13. Solingen Rv, Dullemond K, Gameren Bv. Evaluating the Effectiveness of Board Game Usage to Teach GSE Dynamics. International Conference on Global Software Engineering (ICGSE). Helsinki, Findland: 166-175 (2011)
14. Drappa A, Ludewig J. Simulation in software engineering training. Proceedings of the 22nd international conference on Software engineering. Limerick, Ireland: ACM: 199-208 (2000)
15. Iwai K, Akiyoshi M, Samejima M, Morihisa H. A Situation-Dependent Scenario Generation Framework for Project Management Skill-up Simulator. 6th International Conference on Software and Data Technologies. Seville, Spain: 408-412 (2011)
16. Yilmaz M, O'Connor RV. An Approach for Improving the Social Aspects of the Software Development Process by Using a Game Theoretic Perspective. 6th International Conference on Software and Data Technologies. Seville, Spain: 35-40 (2011)
17. Pelachaud C. Studies on gesture expressivity for a virtual agent. Speech Commun;51(7):630-639 (2009)
18. Bevacqua E, Pelachaud C. Expressive audio-visual speech: Research Articles. Comput Animat Virtual Worlds;15(3-4):297-304 (2004)

19. Kavakli M, Rudra T, Li M. An Embodied Conversational Agent for Counselling Aborigines. 6th International Conference on Software and Data Technologies. Seville, Spain: 371-376 (2011)
20. Corder D, U A. Integrating Second Life to enhance global intercultural collaboration projects. ACM Inroads;1(3):43-50 (2010)
21. Clear T, Daniels M. 2D & 3D Introductory Processes in Virtual Groups. 33rd ASEE/IEEE Frontiers in Education Conference; 2003 5-8 Nov, 2003; Boulder, Colorado: IEEE. p. S1F1-SIF6 (2003)
22. Parvathanathan K, Chakrabarti A, Patil PP, Sen S, Sharma N, Johng Y. Global Development and Delivery in Practice: Experiences of the IBM Rational India Lab: IBM Press, (2007)
23. Casey V. Imparting the importance of culture to global software development. ACM Inroads;1(3):51-57 (2011)
24. Hall W. Managing Cultures: Making Strategic Relationships Work. 1 ed: John Wiley & Sons, (1996)
25. Hofstede G, Hofstede GJ. Cultures and organizations: software of the mind. 2nd ed. New York, NY, USA, (2005)
26. House RJ, Hanges PJ, Javidan M, Dorfman P, Gupta V. Culture, Leadership, and Organizations: The GLOBE Study of 62 Societies. Thousand Oaks, California, USA: Sage Publications, (2004)
27. Trompenaars A, Hampden-Turner C. Riding the waves of culture: understanding cultural diversity in global business: McGraw Hill, (1998)
28. Hall ET. Beyond Culture: Anchor Press, (1976)
29. MacGregor E, Hsieh Y, Kruchten P. Cultural patterns in software process mishaps: incidents in global projects. Proceedings of the 2005 workshop on Human and social factors of software engineering. St. Louis, Missouri: ACM: 1-5 (2005)
30. Cunha ADD, Greathead D. Does personality matter?: an analysis of code-review ability. Commun ACM;50(5):109-112 (2007)
31. Capretz LF. Personality types in software engineering. Int J Hum-Comput Stud;58(2):207-214 (2003)
32. Tsai M-T, Huang Y-C. Exploratory learning and new product performance: The moderating role of cognitive skills and environmental uncertainty. The Journal of High Technology Management Research;19(2):83-93 (2008)
33. Strang KD. Leadership substitutes and personality impact on time and quality in virtual new product development projects. Project Management Journal February 2011;42(1):73–90 (2010)
34. Yoo Y, Alavi M. Emergent leadership in virtual teams: what do emergent leaders do? Information and Organization;14(1):27-58 (2004)
35. Kayworth TR, Leidner DE. Leadership Effectiveness in Global Virtual Teams. J Manage Inf Syst;18(3):7-40 (2002)
36. Yvonne Dittrich RG. Exploring the Role of Instant Messaging in a Global Software Development Project. International Conference on Global Software Engineering (ICGSE). Helsinki, Findland: 103-112 (2011)
37. Wallace RS. The Anatomy of A.L.I.C.E. In: Netherlands S, editor. Parsing the Turing Test: 181-210 (2008)