An Educational Environment for Training Skills for 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— **Global Software Development (GSD) is a recent trend that allows team members to be located on different remote sites, thus forming a network of virtual teams working on the same projects which confront the typical problems caused by distance. The stakeholders involved in the project must be trained to deal with communication difficulties such as those related to cultural and language differences or the appropriate use of groupware tools when English is used as the lingua franca.**

We present a simulator which allows universities and companies to train students or inexpert engineers in the new challenges introduced by GSD, thus avoiding the problems that this activity entails in real settings. Our proposal places non-native English speakers in predefined virtual GSD scenarios in which they will interact with virtual agents of different cultures that play different roles in the project.

Keywords- Global Software Development; Teaching Model; Educational Environment; Simulators; Software Engineering Education; Global Requirements Elicitation; Virtual Agents

I. INTRODUCTION

Global Software Development (GSD) [1] is an emerging trend of the software industry whose main objective is to take advantage of the wider availability of a skilled workforce in decentralized zones, thus allowing team members to be located on various remote sites and forming a network of virtual teams that interact by using collaborative tools.

Problems, caused mainly by distance, frequently appear in these environments: inadequate communications, language and cultural differences, time differences and knowledge management complexity [2]. Communication is less fluid than in co-localized settings because of the high response delays, lack of face-to-face contact and misunderstandings, which may cause frustration and a lack of motivation. One of the main challenges in GSD is to achieve a common understanding and trust between multicultural and multilingual teams [2].

Frequently, engineers are not prepared to confront these new challenges which are not part of their conventional education. In addition, training these skills is a complex task, as it requires preparing learners to deal with people who have different customs, languages, beliefs, skills and ways of interacting [3]. In this sense, learner must know the different cultures involved in the communication in order to avoid conflicts that can appear as a consequence of an inappropriate interaction [4]. The knowledge of a common language and the use of a common terminology are essentials to reach an understanding. However, engineers must also learn to manage uncertainty and ambiguity present in GSD environments. Finally, they must also have experience in the use of knowledge management tools and develop leadership, time management and cognitive skills [4].

The interaction between distributed members requires the use of technology such as e-mail, video conference, wiki and instant messaging [2]. Groupware skills are therefore a key requirement in GSD. However, recently graduated professionals generally lack these skills since they have seldom been involved in real projects [3]. Many strategies found in literature address the teaching and training of students and software engineers in GSD through academic courses [5] or learning environments [6]. However, reproducing the complexity of real environments in universities is difficult since it requires a great deal of coordination between different universities [3]. Furthermore, companies are not always willing to invest time and resources in training programs, which might be too risky in real scenarios.

In this work, we present a simulator that permits active training, with which students can learn by interacting with Virtual Agents (VAs) and can confront typical language and cultural problems, thus avoiding the impact of mistakes in real settings. VAs are characterized by a particular culture and will place learners in GSD scenarios specifically designed to improve their capabilities with regard to cultural and language differences. This will permit self-sufficient training at any time without requiring coordination with other members.

II. GSD SIMULATOR

Our simulator uses VAs that behave like people from different nationalities in order to train learners to confront cultural and language differences through the use of textual dialogues. The VAs have personalities and display emotions, and use text-to-speech capabilities, appearing to care about the students' actions in order to increase their motivation.

VAs can play any role in a GSD project, such as customer, requirements analyst, developer, project manager, etc., thus allowing the student to be placed in a wide range of realistic scenarios. We have, moreover, also designed a VA which always plays the role of Virtual Colleague (VC) or team mate [7] which will help learners to cope with the scenario by guiding them through certain steps.

Our simulator has been based on a client-server architecture. Both the instructors and learners have an interface with which to access the services provided by the central server.

The **instructors' interface** allows managing learners' information and tasks, validating automatic corrections and examining students' actions. It also allows existing training scenarios to be edited and new ones to be created, along with defining new VAs with specific cultures and personalities, thus providing a customizable environment.

The **learners' interface** allows them to execute the available scenarios. Firstly, the VC will present the scenario in which the learner will have a simulated meeting to both the learners and the VAs involved. The VC will guide the users through the scenario and will correct any inappropriate interventions with the goal of allowing the learners to realize what is or is not correct and to learn from their mistakes. After the scenario has been executed, the learners will normally have to fill in certain documents, pass an exam or fill in a questionnaire that can be automatically or semi-automatically evaluated.

The server side provides all the services required by the clients and will manage all the administrative information associated with instructors and learners (exams, grades, learners' status, etc.) as well as the GSD scenarios containing all the information required for their execution and the conversational, cultural and language knowledge associated containing general purpose rules and best practices.

In this sense, **cultural knowledge** is based on the existing literature of Hall [8] and Hosfstede [9] and deals with cultural dimensions for each pair of cultures considering: the use of titles, presentations and greetings, starting and finishing a conversation, motivation and rewards, requests, negotiations, conflicts resolution, etc. The **language knowledge** base contains the rules for all the language pairs that the VC will use in order to correct learners' mistakes, such as those that take place when English is used as a lingua franca [10]: the incorrect use of "false friends", incorrect plural formations, avoidance of passive forms, the absence of the third person, the use of redundant prepositions, the overuse of certain verbs of high semantic generality, etc.

A. *Scenarios definition*

Our scenarios are defined by a VC and one of more VAs (shown Figure 1), that will interact with students by using a chatbot system [11] which will answer the students' questions in the context of a GSD problem, using natural language by simulating stakeholders of different nationalities with different appearances, cultures, gestures and voices.

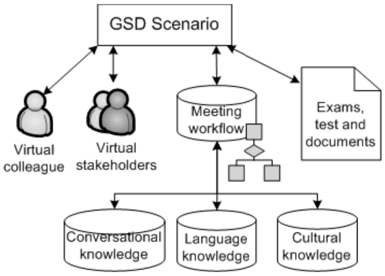


Figure 1. GSD scenario definition

Both the VAs and the VC guide the scenario by concentrating on the *Meeting Workflow* in order to follow a logical sequence during the conversation, in which each phase of the workflow matches a concrete part of the conversation. We also employ decision points that send learners in different directions based on their responses or actions. In this way, we immerse learners in a story in which they can influence its outcome. The information is placed in a context and revealed in a linear fashion, prompting the student to choose between several responses that influence the execution of the scenario. This architecture makes it possible to simulate profound and insightful conversations, avoiding speech repetitions and out of context interventions.

Each phase of the workflow has a concrete *conversational knowledge* and also a context specific *language* and *cultural knowledge*. The phase also contains information about its priority in those cases in which learners can choose a path, and this serves to evaluate their actions.

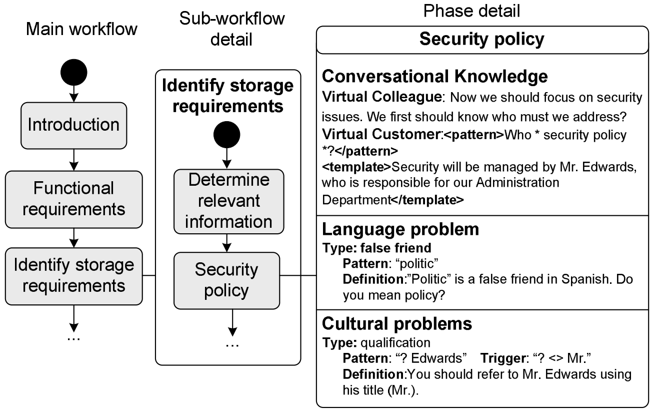
While interacting with the learner, the VC detects cultural problems and the inappropriate use of language by using these knowledge bases, and proposes a more correct manner in which to construct the sentence for that culture. If the learner chooses a wrong path in the scenario, the VC can also help him/her by providing immediate feedback. This helps to reduce the loss of time on incorrect paths, removes unnecessary complexity in the scenario design and prevents the learner from returning to a previous conversation.

III. TEACHING REQUIREMENTS ELICITATION

Since GSD is a highly extensive area, we have focused our initial efforts on the Requirements Elicitation (RE) stage, since this *is a highly communicative process* particularly affected by poor communication, and difficulties regarding cultural and time differences.

We propose a scenario for our simulator in which the learner, playing the role of interviewer, will interact with VAs through several simulated elicitation meetings using English as the lingua franca in order to elicit a set of software requirements. For this purpose, we have prepared the culture and language knowledge of the scenario to include the typical problems that Spanish people have when using English.

In this case, the VC will explain the context of the problem and will introduce the VAs and their culture and role in the project. The scenario continues with the interview, according to our *Meeting Workflow* which is oriented to the elicitation of a set of functional requirements as well as a set of storage requirements. As we show in the example in Figure 2, a *Meeting Workflow* can contain sub-flow charts that guide the scenario in the context of that phase. In this example, the main workflow contains the sub-workflow "identify storage requirements". Each simple phase contains the context-specific conversational knowledge and language and cultural problems that can appear during the interaction. Concretely in the "security policy" phase of the example we see the patters that will use the chatbot for the VA and the VC, and also how we deal with a typical false-friend mistake and how the VC will correct the learner when he/she improperly refers to a person. After completing all the meetings, learners will fill in a requirements document with the requirements detected.



Virtual Colleague: Now we should focus on security issues. We first should know who must we address?
Virtual Customer:**<pattern>**Who * security policy *?**</pattern>**
<template>Security will be managed by Mr. Edwards, who is responsible for our Administration Department**</template>**

Type: false friend
Pattern: "politic"
Definition:"Politic" is a false friend in Spanish. Do you mean policy?

Type: qualification
Pattern: "? Edwards" **Trigger**: "? <> Mr."
Definition:You should refer to Mr. Edwards using his title (Mr.).

Figure 2. Meeting workflow example

IV. Conclusions and Future Work

In this work we have presented a training simulator that introduces learners to GSD activities. The usage of VAs implies many advantages, since they are always available so trainers can work with them at any moment. VAs can simulate the behavior of people from many countries, playing different personalities and thus giving users the opportunity to train in many different scenarios. This simulator helps the students to develop the skills needed when working in GSD, especially regarding to cultural and language differences.

In a preliminary phase we have focused our research on RE meetings. However, in the future we hope to deal with other stages of GSD in which other types of meeting might take place and language and cultural problems might appear (e.g. tracking meetings, negotiation meetings, project initiation meetings, etc.). This will allow us to provide a wide variety of training scenarios dealing with several of the typical situations in GSD, in order to obtain a complete and autonomous training environment that will require a minimum effort on the part of the instructor, thus allowing learners to train at any moment without depending on the availability of other partners or colleagues.

Acknowledgment

This work has been funded by the PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01). It is also supported by the ENGLOBAS (PII2I09-0147-8235) and MELISA (PAC08-0142-3315) projects, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain, and also by the FABRUM project (PPT-430000-2008-063), Ministerio de Ciencia e Innovación, in Spain.

References

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

[2] D. Damian*, et al.*, "Addressing the Challenges of Software Industry Globalization: The Workshop on Global Software Development," *ICSE,* pp. 793-794, 2003.

[3] D. Damian*, et al.*, "Instructional design and assessment strategies for teaching global software development: A framework," presented at the Proceedings of the 28th international conference on Software engineering, Shanghai, China, 2006.

[4] A. Aken and M. D. Michalisin, "The impact of the Skills Gap on the Recruitment of MIS Graduates," in *SIGMIS-CPR'07*, St. Luis, Missouri, USA., 2007.

[5] C. Murphy*, et al.*, "A distance learning approach to teaching eXtreme programming," *SIGCSE Bull.,* vol. 40, pp. 199-203, 2008.

[6] T. Schümmer*, et al.*, "Teaching distributed software development with the project method," presented at the Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!, Taipei, Taiwan, 2005.

[7] C.-Y. Chou*, et al.*, "Redefining the learning companion: the past, present, and future of educational agents," *Comput. Educ.,* vol. 40, pp. 255-269, 2003.

[8] E. T. Hall, *Beyond Culture*: Anchor Press, 1976.

[9] G. Hofstede and G. J. Hofstede, *Cultures and organizations: software of the mind*, 2nd ed. New York, NY, USA, 2005.

[10] J. Grzega, "Reflection on Concepts of English for Europe: British English, American English, Euro-English, Global English," *Journal for EuroLinguistiX 2,* pp. 44-64, 2005.

[11] R. S. Wallace, "The Anatomy of A.L.I.C.E," in *Parsing the Turing Test*, S. Netherlands, Ed., ed, 2008, pp. 181-210.