An implementation architecture for scenario-based simulations

Raja Lala[1], Johan Jeuring[1,2], Jordy van Dortmont[1]

[1]Utrecht University
[2]Faculty of Management, Science and Technology, Open University of the Netherlands

Abstract. The past years have witnessed an increased use of applied games for developing and evaluating communication skills. These skills benefit from interpersonal interactions. Providing feedback to students practicing communication skills is difficult in a traditional class setting with one teacher and many students. This logistic challenge may be partly overcome by providing training using a simulation in which a student practices with communication scenarios. A scenario is a description of a series of interactions, where at each step the player is faced with a choice. We have developed a scenario editor that enables teachers to develop scenarios for practicing communication skills. A teacher can develop a scenario without knowledge of the implementation. This paper presents the implementation architecture for such a scenario-based simulation.

Communication skills are best developed in a realistic setting (Realdon, Zurloni, Confalonieri, & Mantovani, 2012). Scripting different ad hoc perspectives is a prerequisite for a narrative structure to reproduce both the flexibility and regularity of communication. A simulation offers an environment for such a realistic situation.

Utrecht University uses a simulation in communication skills courses. Teachers develop communication scenarios in a web-browser based editor and the resulting scenarios are played in the Communicate! application (Jeuring et al, 2015). The simulation is a one-to-one interactive learning environment (Woolf, 2010) which provides step-wise feedback to a student. It supports goal-based learning-by-doing (Schank, Fano, Bell, & Jona, 1993) of communication skills. The simulation has been tested in practice with Psychology, Pharmacy, Medicine & Veterinary medicine students and city council healthcare first-line support employees**.** Scenario authoring is difficult because a teacher needs to possess pedagogical knowledge, domain understanding and storytelling creativity (Niehaus, Li, & Riedl, 2011). An important aspect of Communicate! is the de-coupling of scenario development by communication skills experts from the implementation. Thus a domain expert may focus on complex scenario creation.

We distinguish three phases in developing and playing scenarios: prepare, play and reflect. The following figure schematically describes our implementation architecture.

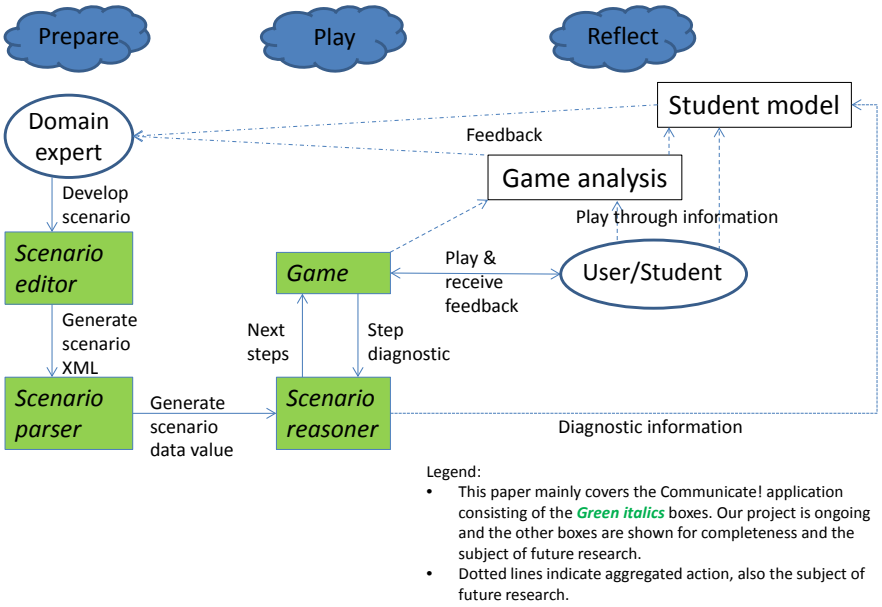

Figure 1: An implementation architecture of scenario-based simulations

In the Prepare phase a communication expert iteratively develops a scenario in the scenario editor as a directed acyclic graph of steps, and specifies the respective scores and feedback per step. Compared to the GIFT framework (Goldberg, Sottilare, & Sinatra, 2015) which offers a talking head with a question-answer natural language interface, we focus on scripted communication scenarios.

The graph represents the pedagogical communication content knowledge of the expert. It is validated against a schema that describes the structure of scenarios. The scenario parser uses the graph to generate a scenario specific reasoner. At run-time the game interacts with the scenario reasoner, which provides information about the possibilities at each step in the series of interactions. Incremental scores and emotion parameters are fed-back by the reasoner to the game. The game user interface shows a virtual character and an appropriate background location, and uses the game logic to present the game to the user/student.

Usability of authoring environments often comes at the expense of expressiveness (Murray, 2003). Our scenario editor tries to combine usability and expressiveness for the domain of communication scenarios. Besides standard sequence, choice, and conditional options, two unique aspects we offer in our scenarios are interleaving (Heeren & Jeuring, 2011) and premature endings. Interleaving is particularly useful when students have to perform multiple (sub)tasks, but the order in which these tasks are performed is not important. Premature endings enable a student to skip the following steps in a sequence. Interleaving and premature endings add expressiveness to the editor, and give the author the possibility to obtain a high-level view of a scenario. The editor is implemented in JavaScript and runs in a web-browser, which makes it easily accessible to domain experts.

The Reflect phase is not directly implemented in the Communicate! game, but under development as an independent component that analyses the play-throughs of students and provides insight into student behavior. Effectivity of scenario development, especially using statistical mechanisms like Cronbach's alpha or RIT (Rasch unit scale) values is also an area for future research.

We compared our editor with four dialogue/scenario editors available in the Unity asset store. These assets range from simple tools without advanced features to advanced tools that need a game-developer to program/simulate the game. One of the primary goals of ITSs is to allow practicing educators to become more involved in their creation (Murray, 2003). Communicate! has been well adopted already, and is used by more than twenty teachers/teaching assistants in the above mentioned domains, and played by over a thousand students.

In conclusion, our implementation architecture for communication scenarios allows domain experts to develop scenarios for practicing communication skills without knowledge of the implementation of the simulation.

Acknowledgements. This work has been partially funded by the EU H2020 project RAGE (Realising an Applied Gaming Eco-system); http://www.rageproject.eu/; Grant agreement No 644187. This paper reflects only the authors' view, the European Commission is not responsible for any use that may be made of the information it contains.

References

Goldberg, B., Sottilare, R., & Sinatra, A. (2015). Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design Through a Community of Practice. *Proceedings AIED 2015*, LNCS 9112, pages 945–945.

Heeren, B., & Jeuring, J. (2011). Interleaving Strategies. *Proceedings MKM 2011*, LNCS 6824, pages 196–211.

Jeuring, J. et al (2015). Communicate ! — a serious game for communication skills — *Proceedings EC-TEL 2015*, LNCS 9307, pages 513–517.

Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. *Authoring Tools for Advanced Technology Learning Environments*, 493–546.

Niehaus, J., Li, B., & Riedl, M. O. (2011). Automated scenario adaptation in support of intelligent tutoring systems. *Proceedings FLAIRS - 24*, pages 531–536.

Realdon, O., Zurloni, V., Confalonieri, L., & Mantovani, F. (2012). Learning communication skills through computer-based interactive simulations. *Emerging Communication: Studies in New Technologies and Practices in Communication*, *9*(c), 276–298.

Schank, R. C., Fano, A., Bell, B., & Jona, M. (1993). The Design of Goal-Based Scenarios. *Journal of the Learning Sciences 3(4)*, pages 305–345.

Woolf, B. P. (2010). *Building Intelligent Interactive Tutors Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.