

Analysing and adapting communication scenarios in virtual learning environments for one-to-one communication skills training

Raja Lala¹, Johan Jeuring^{1,2}, and Timo Overbeek¹

¹Utrecht University

²Faculty of Management, Science and Technology, Open University Netherlands

Abstract. Studies show that pre-game content that is customized to fit a player could lead to better learning than a fixed sequence of worked examples and problem solving. A scenario is a description of a series of interactions between a player and a virtual character for one-to-one communication skills training, where at each step the player is faced with a choice between statements. A communication skills teacher/expert (author) develops a scenario in an authoring tool. A simulation presents such a scenario to a player. In this paper we apply the concept of code-smells [2] to a scenario. We define a scenario-smell as a symptom of a scenario that could be an indicator of an error or incorrect design. We use scenario-smells to evaluate a scenario pre-game. A scenario author can use an evaluation to assess and adapt a scenario to better train a player in a game/simulation. This paper presents work in progress.

1 Introduction

Many university and vocational programs train students in communication skills. Communication skills are best learned through practice, in role-play or with a simulated patient [1]. In a digital learning environment for training communication skills, a student often performs a conversation with a virtual character, and the learning environment assesses the performance of the student against learning goals for the conversation.

A student's communication skill-level varies. In a survey of Adaptivity Challenges in Games and Simulations [6], Lopes et al postulate that serious games and simulations have to become more challenging, unpredictable and player-centric, to be fully embraced as an effective way of knowledge transfer. They survey literature and games with respect to adaptivity and conclude that research focuses primarily on adaptive game mechanics, AI, NPCs, and narratives. They distinguish two adaptivity methods: on-line (or in-game) using a player model(s) to control the adaptation of NPC run-time behaviour and off-line (or pre-game) content that is customized to fit a player. An on-line method is predictable to a certain degree, as such a method requires a priori creation of variations. They also find that fewer research groups focus on off-line adaptive scenarios.

Najar et al [8] present a study that compares student performance in a fixed sequence of worked examples and problem solving with an adaptive strategy that determines a task (worked example/problem solving/faded example) based on the amount of assistance needed by a student. Their results show that a student in an adaptive condition learns more than their peers presented with a fixed sequence of worked examples and problem solving.

A scenario is a description of a series of interactions between a player and a virtual character for one-to-one communication skills training, where at each step the player is faced with a choice between statements. A communication skills teacher/expert (author) develops a scenario in an authoring tool. A simulation presents such a scenario to a player. In this paper, we apply the concept of code-smells [2] to a scenario. A code smell refers to a common symptom of a particular error in a piece of code. For example, the occurrence of duplicated code in a program is a code smell, and often duplicated code can be removed. A code smell is not necessarily an error, but a symptom of a potential quality problem. Similarly, a scenario-smell is a symptom of a potential problem in a scenario.

This paper is organised as follows. Section 2 discusses two main types of virtual learning environments that offer scenario based assessment and training. Section 3 describes the Communicate! authoring tool. Section 4 discusses challenges in scoring a scenario and presents assumptions to evaluate a scenario. Section 5 presents the work in progress, future work and conclusions.

2 Virtual learning environments for communication skills training

Some learning environments, such as Enact [7] and deLearyous [11], offer game-play tailored to a specific psychological model. Enact [7] is based on five styles of handling interpersonal conflict proposed by Rahim [10]. Enact and similar serious games where the game-play is intrinsically bound to a specific psychological model are hard to evaluate or adapt to a new psychological model or protocol.

Other learning environments, such as Communicate! [4] and Visual Scene-Maker [3], provide an authoring tool for developing scripted dialogues. A domain expert such as a communication-skills teacher develops a scenario, including a dialogue and score(s) on a learning goal(s) in these authoring tools. A user/player plays a scenario in a learning environment and assesses her performance on the learning goals. This approach allows domain experts to develop various kinds of communication scenarios. These scenarios often follow a protocol, for example a doctor delivering bad news to a patient, or a pharmacist providing medicines to a first-time user.

Communicate! is used by more than twenty teachers/teaching assistants at Utrecht University [5] since three years. The authoring tool provides a clear separation between a scenario author and the game/simulation environment. A scenario author develops scenarios for practising communication skills without

knowledge of the implementation of the simulation, and does not need programming skills. To find out how scenario authors experience the authoring environment, we interviewed six scenario authors [9]. In the interviews, the authors expressed a clear need for scenario evaluation.

3 Authoring a scenario in Communicate!

A scenario in Communicate! [5] is a sequence of interleaved subjects (see Fig. 1), where a subject is a dialogue between a player and a virtual character. A scenario author models a communication protocol from top to bottom. Subjects at the same horizontal level are interleaved and a player gets statement choices from these interleaved subjects with no predetermined order in a simulation.

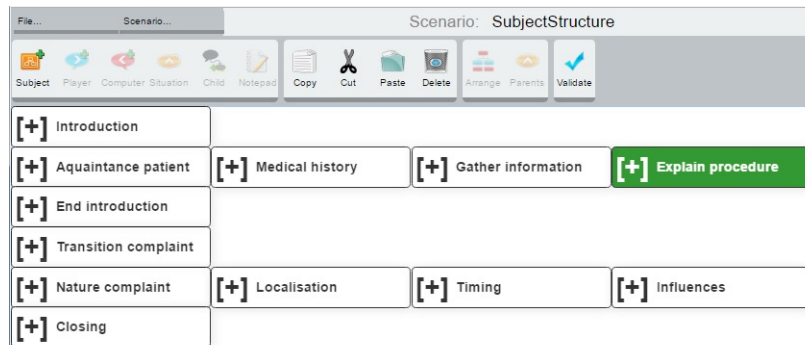


Fig. 1. Example Interleaving subjects in Communicate!

A dialogue is represented as a directed acyclic graph. Player and virtual character statements are nodes and the flow of conversation is an edge, see Fig. 2. A player statement usually has an incremental score, emotional effect on an NPC, and feedback text for a player. Multiple player nodes per computer node indicate multiple statement choices for a player.

A scenario author can further increase variability in a dialogue by marking a node with the following:

- Conditional: show only if the condition(s) is met.
- Jump: allow to jump from this node to another node in a subject at an interleave level.
- Early end of subject: allow to jump from this node to a node in one of the subjects at the same horizontal (interleave) level. If there are no other subjects at the same level, allow to jump to a node in a subject in the following vertical (interleave) level.
- End of scenario: terminate a scenario after this statement.

During game-play, a player receives statement choices at each step in a series of interactions with a virtual character. These statement choices depend on a choice of a player and how a communications teacher scripted the scenario using the features described above. At the end of a simulation a player receives a final score on the learning goals of the scenario.

4 Evaluating a scenario

4.1 Challenges in scoring

A major challenge for a communications teacher is how to score a scenario in terms of learning goals [9].

For a single statement it is usually not hard to distinguish good and bad choices. Spreading a learning objective score across nodes in a graph is more challenging and can lead to inadvertent effects. Fig. 2 shows two examples. The left graph depicts a graph with a single scoring parameter, the right graph shows the same graph with two parameters. We show the delta scores for each of the player statement choices marked r1 to r6.

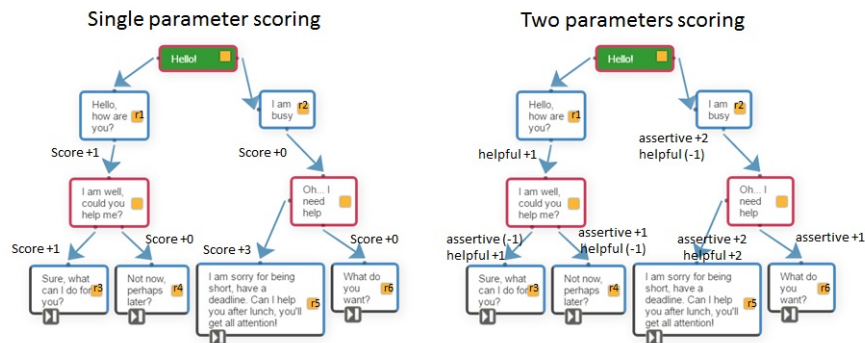


Fig. 2. Example of scoring challenges

In the left graph, a player chooses between r1 and r2 in the first step of a simulation, where r1 yields a better score. If a player chooses this option, she gets the choices r3 and r4 in the following step. A natural assumption is that choosing a statement with the highest score at each step in a dialogue results in a maximum score for a scenario. The sequence of choices r1r3 yields a total score of +2. However, the sequence r2r5 has a total score of +3, the best score for this scenario. The scenario author might have made an error, or made an explicit decision, but the situation ‘smells’.

Scoring complexity increases with more parameters, as shown in the right graph. In this graph, the best scores are the sequences of r1r3 (yields assertive: -1, helpful: +2), and r2r5 (yields assertive: +4, helpful: +1). Note that playing the same scenario could lead to a 'helpful' or an 'assertive' player-profile. This might be expected behaviour, but scoring this scenario is more involved than with a single parameter. Scoring complexity further increases at the subject level with the use of interleaving and other features mentioned in Section 3.

4.2 Analysis of a scenario

We apply the concept of scenario smells to evaluate a scenario by making assumptions about scoring. The basis of a dialogue is a directed acyclic graph. Like the concept of a code-smell that extracts code patterns as a smell, we extract graph information and link it to scoring and other scenario characteristics to detect scenario-smells.

A common assumption is that the longest path in a dialogue graph yields a maximum score. The longest path contains most nodes and therefore has the best chance to maximize the score. Interestingly, we could also assume that the longest path does not yield the highest possible score. A player taking a longer path is perhaps inefficient. In an evaluation we report whether or not a longest path yields a maximal score.

Another common assumption is that the shortest path does not yield a maximum score. Many dialogues have at least one obstacle, such as an unwilling NPC or a bad statement choice. If a player fails to address this obstacle, the dialogue ends prematurely. This shortest path should not result in a maximum score. Sometimes, however, a more direct (shorter) approach is better, for instance in a bad news conversation. The scenario smell of the shortest path versus the maximum score should match the expectation of the scenario author.

We assume that both the minimum and maximum score of all parameters are attainable in a scenario. For example, if an author specifies that the values of a parameter are in between 1 and 10, a player might assume that 6 corresponds to a sufficient result; 8 to a good result and 10 to a perfect result. If it is impossible to achieve a score higher than an 8, then a "perfect" performance (of 8) does not correspond to the perceived result.

An interesting aspect is the relationship between parameters. For instance, if two parameters have a strong positive correlation, we could advise a scenario author to combine these parameters; since having two different parameters might be superfluous. This is conceptually similar to the duplicated code code-smell. Some parameters, for example for conflicting goals as in the right graph of Fig. 2, should have a negative correlation. The sign and degree of correlation for parameters in a scenario should match the expectations of a scenario author.

On a subject level with interleaving, Communicate! allows a player to traverse interleaved subjects in any order. We assume that a score of a path within a subject is directed related to an overall score of a scenario. If a player takes a different path in a subject that yields a higher (or lower) score, then the overall score of the player for this scenario should also be higher (or lower).

We call a scenario that satisfies this property subject-monotone. Some subjects are required to be subject-monotone, some not. We can check whether or not subject-monotonicity holds for a scenario, and report this to the scenario author.

Some assumptions relate to graph characteristics and not to scoring parameters. Examples are: all nodes should be reachable; nodes should not connect back to a parent; every path should have an endpoint. Most of these properties are already checked in the current Communicate! editor; except the requirement that all nodes are reachable, which we will include in our evaluation-tool.

Summarising, our evaluation tool reports the following scenario smells:

- Always choosing the best option does not result in the maximum score.
- A longest path does not yield the maximum score.
- A shortest path yields the maximum score.
- The minimum and maximum score of all parameters is not attainable.
- It is possible to obtain a parameter score higher (lower) than the specified maximum (minimum) score for the parameter.
- Two parameters correlate.
- A scenario is not subject-monotone.
- Some nodes are not reachable.

5 Work in progress, future work and conclusion

5.1 Work in progress and future work

We have started with an implementation of an evaluation tool that detects the scenario-smells described in subsection 4.2 in a scenario. We report the result of a run of the evaluation tool in a report to a scenario author. A scenario author can use this report to annotate and/or modify a scenario.

A subject of future research is to perform post-game analysis on a scenario. This involves analysis of, for example a player/student(s) play-through, a statement/question(s) in a scenario that students struggle with, a scenario's contribution to learning etc.

5.2 Conclusion

Communication skills are needed in the majority of professions and are best learned through practice. These skills can be developed in virtual learning environments that simulate real-world communication scenarios, for instance a bad news conversation.

We investigate pre-game methods to evaluate a scenario from the Communicate! authoring tool and develop assumptions to analyse a scenario. We have an implementation to calculate maximum score paths in a scenario and we plan to further implement assumptions to determine scenario-smells.

We aim to provide an analysis tool for a scenario author, usually a non-programmer, to adapt a scenario to assess a student in a communication skill(s). The result of an assessment can be used to recommend an adapted scenario(s) depending on the personalised needs of a student.

6 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 author's view, the European Commission is not responsible for any use that may be made of the information it contains.

References

1. Marianne Berkhof, H. Jolanda van Rijssen, Antonius J.M. Schellart, Johannes R. Anema, and Allard J. van der Beek. Effective training strategies for teaching communication skills to physicians: An overview of systematic reviews. *Patient Education and Counseling*, 84(2):152–162, 2011.
2. Martin Fowler and Kent Beck. *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 1999.
3. Patrick Gebhard, Gregor Mehlmann, and Michael Kipp. Visual scenemaker—a tool for authoring interactive virtual characters. *Journal on Multimodal User Interfaces*, 6(1):3–11, 2011.
4. Johan Jeuring, Frans Grosfeld, Bastiaan Heeren, Michiel Hulsbergen, Richta IJntema, Vincent Jonker, Nicole Mastenbroek, Maarten van der Smagt, Frank Wijmans, Majanne Wolters, and Henk van Zeijts. Communicate! — a serious game for communication skills —. In *Proceedings EC-TEL 2015: Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning*, volume 9307 of LNCS, pages 513–517. Springer, 2015.
5. Raja Lala, Johan Jeuring, and Jordy van Dortmont. An implementation architecture for scenario-based simulations. In *Proceedings ITS 2016: 13th International Conference on Intelligent Tutoring Systems*, volume 9684 of LNCS, pages 531–533. Springer, 2016.
6. Ricardo Lopes and Rafael Bidarra. Adaptivity challenges in games and simulations: A survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(2):85–99, June 2011.
7. Davide Marocco, Daniela Pacella, Elena Dell'Aquila, and Andrea Di Ferdinando. Grounding serious game design on scientific findings: The case of enact on soft skills training and assessment. In *Proceedings EC-TEL 2015 Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning*, volume 9307 of LNCS, pages 441–446. Springer International Publishing, 2015.
8. Amir Shareghi Najjar, Antonija Mitrovic, and Bruce M. McLaren. Learning with intelligent tutors and worked examples: selecting learning activities adaptively leads to better learning outcomes than a fixed curriculum. *User Modeling and User-Adapted Interaction*, 26(5):459–491, 2016.
9. Timo Overbeek, Johan Jeuring, and Raja Lala. Using scenario smells to analyse scripted communication scenarios in virtual learning environments. Technical Report UU-CS-2017-008, Department of Information and Computing Sciences, Utrecht University, 2017.
10. M Afzalur Rahim. A measure of styles of handling interpersonal conflict. *Academy of Management journal*, 26(2):368–376, 1983.

11. Jeroen Wauters, Frederik Broeckhoven, Maarten Overveldt, Koen Eneman, Frederik Vaassen, and Walter Daelemans. delearious: An interactive application for interpersonal communication training. In *Serious Games: The Challenge: Joint Conference of the Interdisciplinary Research Group on Technology, Education, and Communication, and the Scientific Network on Critical and Flexible Thinking Ghent*, volume 280 of *Communications in Computer and Information Science*, pages 87–90. Springer, 2012.