Improving Serious Game Design with Collaborative Storytelling

Casper Harteveld, Stephan Lukosch, and Rens Kortmann

Delft University of Technology, Faculty of Technology Policy and Management,
Jaffalaan 5, 2628BX Delft, The Netherlands
{c.harteveld,s.g.lukosch,l.j.kortmann}@tudelft.nl

Abstract. It is challenging to design a serious game. In addition to being fun, such a game needs to be valid and meaningful. For this reason, requirement and domain knowledge elicitation are more important in designing serious games compared to designing entertainment games. Despite this importance, the elicitation occurs frequently in an unstructured manner. In this paper, we propose to use and embed collaborative storytelling in the serious game design process. We expect that this improves the quality of the game as it helps game designers to better understand the domain for which they design it. Furthermore, it will allow clients, users, and subject-matter experts to participate in and learn from the design process. We reflect on the use of collaborative storytelling by describing how an actual storytelling tool could have been used during the game design process of an existing serious game.

1 Introduction

Designing serious games poses a different challenge than designing entertainment games [1]. With serious games we refer to those games that aim to achieve an effect beyond the game itself. This can be for educational, for marketing, for research, or for any other "serious" purposes. To develop such games successfully, designers need to create a fun, valid, *and* meaningful game. In contrast to this, most entertainment games solely need to be fun. For being able to develop a successful serious game, we outline in this paper a new method to improve the design process, namely *collaborative storytelling*.

To elaborate on the usefulness of this new method it is important to understand that to achieve a valid and meaningful game, developers need to translate a part of reality into a game, while making sure the game actually serves the purpose it is designed for. This requires a close collaboration with clients, users, and subject-matter experts for two important reasons. First of all, they provide the needed requirements and/or domain knowledge. Subsequently, their input is necessary to judge whether the requirements and domain knowledge have been implemented accordingly.

Despite that the elicitation of requirements and domain knowledge is essential to design a good quality serious game, no structured approach exists in practice

that deals with this. To overcome this issue, "stories" can play a (more important) role. Telling stories are not only a given phenomenon of human practice (including games); it is purposefully used as a method or a procedure in different areas of application under the designation *storytelling*. A specific and potentially useful method for serious game design is collaborative storytelling. This aims at the development of a common understanding within a group by coordinated narrating activities (when each person contributes his or her own knowledge and his or her own interpretation of a common experience), to make implicit knowledge explicit. A further special quality in collaborative storytelling that makes it promising for serious game design arises from the restriction on verbal telling of stories, i.e. from the restriction on auditive production and perception.

Based on these potential advantages, we propose in this paper to structure the design process of serious games by using and embedding a collaborative storytelling approach. We expect that the use of collaborative storytelling improves the quality of the game for two possible reasons. Storytelling will help game designers to better understand the domain for which they design a game. Furthermore, storytelling will allow clients, users, and subject-matter experts to participate in and learn from the design process.

In the following, we first further discuss the game design process of serious games. Then, we introduce a tool for collaborative audio-based storytelling. In the following section, we reflect on how the design of a serious game could have benefited from collaborative storytelling. Finally, we discuss our approach and give an outlook on future work.

2 Designing a Serious Game

To understand the need for a systematic approach to elicit requirements and domain knowledge, we outline in this section the design process of a serious games and its issues. First, in Section 2.1, the value of elicitation within a game design process is indicated. Secondly, in Section 2.2, a number of issues are discussed based on the experiences of designing a serious game called *Levee Patroller*.

2.1 The value of elicitation

Although designing a game is largely a creative process, it is possible and also beneficial to formalize, and thus structure, the game design process into specific design steps that need to be taken. Based on widely accepted approaches in game design [2, 3] and software engineering [4], Kortmann and Harteveld [5] described that designing a (serious) game consists of four sequential design phases (see Figure 1). The four phases are explained in order below:

SCOPE In the scoping phase of a game design process, the designer and client determine the aim, the required resources, and the planning of the project.
DESIGN In the design phase, the requirements of the game are elicited and a

functional design of the game is drawn up. This design includes the construction of the system model upon which the game is based.
BUILD In the build phase, a game artefact is constructed. This could be a computer game, but may also be a board game, or a set of rules of a rather abstract game.
TEST In the test phase, developers and the client verify and validate the game. For this they could use test players, or they could evaluate the game themselves; this choice depends on the maturity level of the game under construction.

The authors further argued that the design process should have three decision moments throughout the process [5]. These are necessary to potentially return to one of the preceding phases. This iterative character makes it possible to gradually improve the design. The design team can start the iteration cycles by constructing and evaluating very simplified versions of a game. Over time more complex and complete versions of this game are developed in the iteration cycles that follow.

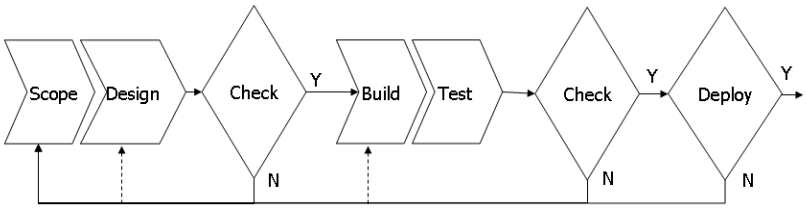

Fig. 1. Game design process by Kortmann and Harteveld [5]

Based on this design process, we can see that elicitation is needed in every phase and every decision moment. In the first two phases it is necessary to elicit requirements from clients and users. With these requirements, the designers are able to set the scope and determine the functional design. During the third phase, the build phase, elicitation of domain knowledge takes place. If the wanted domain knowledge is simple and clear-cut it may not be needed to talk to subject-matter experts. However, if the game is about complex processes or implicit procedures, detailed domain knowledge is desirable. Collaboration with subject-matter experts is further applicable when the domain knowledge itself is ambiguous, undecided, dispersed, or when experts have different views on the subject. In the last phase, the test phase, and at each of the decision moments, it is necessary to elicit feedback to evaluate and validate the design choices. From this, it can be judged whether the requirements and domain knowledge have been implemented accordingly.

The elicitation of these types of information is essential to serious game design as functionality (in terms of validity and meaning) is a critical success factor. Otherwise a serious game may not reach a certain effect beyond the game that

relates to an activity in reality [1]. The problem with serious game design is that the elicitation is largely unstructured. The case of *Levee Patroller* will illustrate a number of issues that may arise due to this lack of systematic inquiry.

2.2 The design of Levee Patroller

Levee Patroller can be described – in game jargon – as a "single-player 3-D first-person game." This means the game is solely played by one user from the perspective of the player character. This game the commercial game engine "Unreal Engine 2." In the game the player's role is that of a "levee patroller." These are people who inspect the levees regularly or in cases of emergency. This inspection activity is of high importance to the Netherlands due to the high risks involved in a possible levee failure.[1] Therefore, it was desired that these professionals can practice their recognition and reporting skills of failures in a safe and virtual environment. This is in particularly important, since it is difficult to do this in reality. These types of failures occur rarely. The basic purpose of the game is to find every virtual failure and report it (see Figure 2).

Fig. 2. A levee failure in *Levee Patroller*

From the original game design process several interesting issues can be noted that could have been improved if a more systematic elicitation method would have been used:

[1] Levees (or dykes) are barriers that protect the land from flooding.

1. **Diversity** The game had a number of clients. These clients differed in vocabulary, organizational setup, and in the type of failures they deal with amongst many other differences. This made it incredibly difficult for the designers to judge to what extent the clients were talking about the same issues on the one hand, and to determine, on the other hand, how they needed to translate the differences to the game.
2. **Lack of or implicit knowledge** The game design heavily depended on the input of subject-matter experts. Little is known about failures. Only a few pictures and reports are available. The knowledge of these experts is largely implicit. This also became clear during the process. Frequently, designs had to be changed, because the visualization was not exactly what the expert had in mind. When the visualizations were made, however, much information about the motivations of the design were lost. The designers could not explain how the virtual failure evolved throughout the process. The design would have highly profited from a way to capture the knowledge of experts and make it explicit.
3. **Dissensus** Another issue related to the subject-matter experts is that amongst each other they did not reach a consensus about the visualization of many failures. Due to time pressure, and an inability of experts to assist the design at all times, the design team mostly decided on pressing issues themselves. If beforehand, it would have been possible to let experts comment on each other in a flexible way, the pressing issues may have been decided on in a more correct manner.
4. **Documentation** In general, this project also had trouble in documenting design choices and feedback from clients and users. Due to this, it has become difficult to trace back why certain decisions were made.

From the above, it becomes clear that the *Levee Patroller* project suffered from an unstructured approach to elicit information and implement this into a game. Valuable information has gone lost to (a) improve the game, (b) give feedback to the users, and (b) discuss with experts why certain design choices were made. Therefore, we argue that this design process could have benefited from a more systematic way of gathering information about the requirements and the domain knowledge. When it comes to this sort of elicitation, we expect that collaborative storytelling can play a role. The next section discusses how a particular tool for collaborative storytelling can be used to realize this.

3 CASTing – A Tool for Collaborative Audio-based Storytelling

Quite recently a comeback of listening can be determined on the basis of the risen demand for audio books [6] and from the unprompted rise of podcasting [7]. In our approach to requirements and domain knowledge elicitation, validation and evolution we focus on oral narrating activities. Spoken language is the basis for telling stories and an essential and quite natural part of human communication.

Verbal telling of stories connects to the prevalent forms of data collection in user-centered requirements elicitation that are interviews and group discussions with stakeholders.

Based on these premises, a special quality in collaborative storytelling arises from the restriction on the auditive production and perception of stories. The focus on oral narrating activities aims at lowering the threshold for people to participate in the process in order to facilitate equal discourse (peer-to-peer) as well as the mutual exchange of the social roles of layman and experts (bottom-up). It is obvious that the use of collaborative storytelling that is aligned to equal discourse or the qualification for participation makes special demands on supporting collaborative information systems. These must be easily accessible, self-describing and conducive for learning; in short, ease of use is essential.

CASTing [8] is an information system that enables audio-based collaborative storytelling in distributed settings. CASTing is based on a suitable collaboration process and offers a client application that allows different actors to collaboratively create audio-based stories as well as a web portal that serves as a community platform to access, review, and rate audio-based stories.

Considering the notion of telling and re-telling stories collaboratively in order to develop a common understanding within a group, it is obvious that users want to report on current events from different perspectives and to comment existing recordings and stories. Users want to place their comments directly at the point of interest and not at the end of an audio recording. In order to avoid cutting the audio recordings and thereby creating a huge bunch of smaller audio recordings, CASTing supports users to set marks in the audio recordings which can be used to link to a section in another audio recordings. Thereby, users can collaboratively specify stories using links between the different audio recordings. The assembly of single audio recordings results in a directed graph representing different alternative stories. By choosing a starting node within this graph, alternative paths, i.e. different versions of a story, can be selected for publication.

Figure 3 shows the publishing perspective of the CASTing user interface. The upper right corner shows a story graph with a highlighted path. The single elements of the path are shown on the left side of the user interface and allow users to navigate within the selected story. Stories chosen for publication are subject to a rating within the group working on a story. Only by this collaborative assembly the alternative representation of scenarios, use cases, stories or requirements becomes possible as a genuine collaborative act of telling stories, in which a group of stakeholders agrees stepwise on a common view of things.

Often requirements and domain knowledge elicitation, validation and evolution are field work and require data collection on site as well as negotiations in distributed settings. Thus, users are collecting audio clips for a shared story located in different locations and possibly with intermittent access to the Internet. CASTing supports this kind of nomadic work and after performing local changes, users are able to synchronize their changes with the latest versions of the group's items and resolve any conflicts that might have occurred.

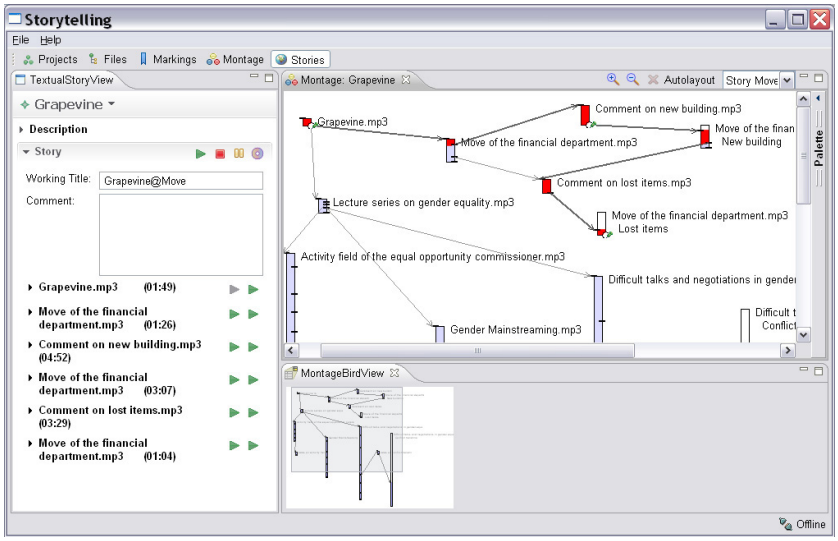

Fig. 3. The publishing perspective

These key concepts distinguish CASTing from other tools for collaborative and multimedia storytelling like StoryMapper [9], TellStory [10], PhotoStory [11] or MIST [12]. Based on these concepts CASTing supports a storytelling process that can be utilized in requirements engineering:

1. Creating a project team
2. Adding audio recordings
3. Segmenting audio recordings
4. Linking audio recordings
5. Publishing a story

These five steps may be executed in any order. It is possible to skip steps or to return to steps in the process. Thereby, CASTing supports the self-regulated collaborative development of a story representing knowledge about a specific domain. Furthermore, this open collaboration process allows all stakeholders to add their understanding to the story graph and thereby support a evolutionary knowledge acquisition as well as reflection process.

As mentioned before, CASTing also offers a web portal which serves as community platform and allows users to register, to create a project, to invite project members, to join ongoing projects, to upload and share audio recordings, to communicate via chat or message board, to view who else is currently, and to review all project-related information. Apart from the above functionality, the web portal also supports users in starting story-related discussions, commenting stories, tagging content and voting on stories. These votes can be used by a project team to decide which of the alternative threads in the story graph is finally published as podcast to all members of the web portal and thus is available to the public.

Such a published podcast can of course be included in a new story project, segmented by marks, linked with other audio recordings, and finally be published again.

4 Reflecting on the design of Levee Patroller

We first described why a serious game design process needs to have more structure for eliciting requirements and domain knowledge. Subsequently, we discussed CASTing, a particular tool that makes collaborative audio-based storytelling possible, and how it can be used in a game design process to deal with the mentioned elicitation issues. Now we want to illustrate how the design of *Levee Patroller* could have benefited from the use of CASTing.

First, at the beginning of the project a web portal would have been published that could be accessed by clients, users, and subject-matter experts. Three separate folders would have been made: requirements, failures, and use. The first folder, requirements, relates to what the game needs to become and have. The second, failures, is concerned about the stories of the possible failures that could be implemented in the game. Finally, the third, use, is about the experiences of users in playing the game. The latter could be used to validate the game.

Due to the distributed setup of CASTing it would have just been a matter of giving some input to the community to enable them to respond by uploading and sharing their audio recordings. This whole process only needs to be instigated once by the designers and further facilitated. A possible input could be the name or a picture of a failure. In case of requirements, it could be a single question, such as "what does a game to train levee patrollers need to have?." If user participation is high, this process requires less effort and leads to much more and better results. By means of voting, users can determine what requirements and failures are more important.

In this situation, experts can comment on each other and possibly reach consensus. Furthermore, clients may get an understanding of each other's position and find a possible middle ground that satisfies each of them or find out that they are actually talking about the same thing. But most importantly, everything is documented. This enables designers to have a coherent and consistent overview of all the feedback provided. It also makes it possible to go through the evolution of the project. An additional (and unexpected) benefit of the system could be that the input of the users can even be used inside the game environment. An explanation of an expert could be implemented as a sound element that is triggered whenever an explanation of a failure needs to be given.

From this, it becomes clear in what way CASTing can be applied for serious games. Although many more applications can be thought of, such as using it actually during the game and at the end during the debriefing, its potential lies foremost in capturing and documenting knowledge that is largely unavailable to laymen, which game designers are when they step into a field that is unknown to them.

5 Conclusion

Designing serious games is challenging. This especially accounts for capturing and eliciting requirements and domain knowledge. These are the essential ingredients to make a game valid and meaningful. To achieve this, collaboration is needed, not only between the designers, clients, users, and subject-matter experts, but amongst the whole community that is involved with the game. Another pressing issue is to make sure this happens systematically. Many serious game projects are too unstructured.

In this paper, we proposed collaborative storytelling to overcome the elicitation issues in serious game design. This method aims at the development of a common understanding within a group by coordinated narrating activities. This way, implicit knowledge can be made explicit, a restriction is made on auditive production and perception, which has several advantages over written production and perception, and people are able to reflect and elaborate on each other. To illustrate what it means to use collaborative storytelling, we discussed one particular tool called CASTing and related this to the game design process of a serious game called *Levee Patroller*. From this, it could be distilled that the original design process could have benefited from collaborative storytelling.

We want to stress that this approach may not be suitable for every type of serious game. Although documenting information is never problematic in itself, the approach can be quite cumbersome. This may especially be true for those serious games that are simple and clear-cut about their purpose and subject matter. This means that project initiators have to carefully see to what extent they can profit from this approach.

To be able to make this trade-off, it is important to actually use collaborative storytelling in practice. Any trade-offs for using the approach may become clearer after that. Another future work that lies ahead of us is to see in what other ways the approach can be used. It may, for example, also have a high potential for usage during the game and at the debriefing.

References

1. Harteveld, C., Guimaraes, R., Mayer, I., Bidarra, R.: Balancing play, meaning and reality: the design philosophy of levee patroller. Simulation & Gaming (forthcoming)
2. Duke, R.: Gaming: the future's language. SAGE (1974)
3. Duke, R., Geurts, J.: Policy games for strategic management: pathways into the unknown. Dutch University Press (1974)
4. Wysocki, R.: Effective software project management. Wiley (2006)
5. Kortmann, R., Harteveld, C.: Agile game development: lessons learned from software engineering. In: Learn to game, game to learn, Proceedings of the 40th conference of the international simulation and gaming association. (2009)
6. Philips, D.: Talking books: The encounter of literature and technology in the audio book. Convergence **13**(3) (2007) 293–306
7. Hein, G., Jakuska, R.: Podcast industry. iLabs – Center for Innovation Research, School of Management, University of Michigan-Dearborn (April 2007)

8. Lukosch, S., Klebl, M., Buttler, T.: Facilitating audio-based collaborative storytelling for informal knowledge management. In Briggs, R.O., Antunes, P., de Vreede, G.J., eds.: Proceedings of the 14th Collaboration Researchers' International Workshop on Groupware (CRIWG 2008). LNCS 5411, Springer-Verlag Berlin Heidelberg (2008) 289–304
9. Acosta, C., Collazos, C., Guerrero, L., Pino, J., Neyem, H., Motele, O.: StoryMapper: A Multimedia Tool to Externalize Knowledge. In: Proceedings of the XXIV Conference of the Chilean Computer Science Society, IEEE CS Press (2004) 133–140
10. Perret, R., Borges, M.R., Santoro, F.M.: Applying group storytelling in knowledge management. In: Groupware: Design, Implementation, and Use, 10th International Workshop, CRIWG 2004. LNCS 3198, Springer-Verlag Berlin Heidelberg (2004) 34–41
11. Schäfer, L., Valle, C., Prinz, W.: Group storytelling for team awareness and entertainment. In: NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction, New York, NY, USA, ACM Press (2004) 441–444
12. Spaniol, M., Klamma, R., Sharda, N., Jarke, M.: Web-based learning with nonlinear multimedia stories. In: Advances in Web Based Learning – ICWL 2006. LNCS 4181, Springer-Verlag Berlin Heidelberg (2006) 249–263