Playing with Narratives

Anke Dittmar

University of Rostock

Department of Computer Science

anke.dittmar@uni-rostock.de

ABSTRACT

Education and play are two different contexts of action which need to be interrelated in order to convey knowledge and skills to students but at the same time to enable them to explore unknown paths. The paper presents a small example study with technology-oriented students who ‘played’ with narrative approaches as used in HCI work analysis. It is shown that their self-reflection by using stories also increases their appreciation of unfamiliar design perspectives. The example illustrates the potential which is offered by playing with HCI education to support an intertwined vertical and horizontal development of students and to better prepare them for multidisciplinary work.

Keywords

Narratives in HCI, play, self-reflection, HCI education.

1. INTRODUCTION

“Play is enjoyed by all and in terms of HCI education it is a concept that can be considered from many viewpoints” [3]. The first question we can ask is why play is enjoyed by all? Play allows people to deal with their environment in an open and carefree way. They are not forced to “accept” predefined meanings of situations but are encouraged to use artifacts as “toys” for finding self-motivated, enjoyable actions. Hence, play may support specific forms of learning.

- Players acquire skills and knowledge in a creative process.
- Play is social. Players experience themselves by reflecting the effects of their actions on the environment and other players.
- Play supports a broader understanding and empathy. It may also support the willingness to change attitudes and habits outside the playing context.

In other words, play is important for adaptation and for renewing life. If people lost their interest in playing their culture would not survive. Though both play and education support learning they have to be considered as two different contexts of action. Education is a process with predefined learning goals. Educators are expected to “control” the achievement of these goals. Additionally, education has a public selection function which is supported by marks or other evaluation schemes. Although play cannot be exploited pedagogically (let alone controlled by marks)

interdependencies between education and play are necessary.

How can HCI education and play be related to each other? In HCI, designers are concerned with concepts such as “fun of use” which need an understanding of play and the ability to play. But play can also be useful in the dialogue between users, designers, programmers and other stakeholders. HCI is committed to multidisciplinary science. Hence, HCI education is not only concerned with training of HCI experts. It also aims at broadening the view of students in other, e.g., technology-oriented fields. Common forms are undergraduate courses or specific courses like usability engineering or user-interface design courses [4]. However, survey courses and courses which emphasize the engineering perspective again may tend to reproduce ‘traditional’ practices with a technological focus.

This paper describes students of computer science and business informatics playing with narrative approaches as used e.g. in HCI work analysis. We show that the appreciation of complementary design approaches may be supported by playing with them and draw conclusions for appropriate conditions in educational settings.

2. AN IDEA EMERGES

Thomas looked around. The tutor and the students of the other group had left the room. Now they could start their first project meeting. He looked at the half asleep Leo. “Hey! When did you go to bed last night?!” Nothing but unrecognizable muttering. Thomas remembered that his friend was still online when he switched off his computer at midnight. He sighed and turned his attention to the others. “Do you take the minutes, Armin?” he asked the guy sitting opposite him. Armin nodded and looked for a biro and paper. “Let’s start!” Thomas tried to appear enthusiastic. „Okay, let’s start“, Gregor agreed. „The topic of the project seems to be quite clear. I think we must first decide which platform and programming language we want to use.”...

Thomas, Leonard (“Leo”), Gregor and Armin are students of computer science in their third semester. They are friends but Thomas and Leo are close friends since school days. They are supposed to work together on a small project as part of their software engineering (SE) course.

3. PLAYGROUND

Thomas, Leo, Gregor and Armin are the main characters of a story developed within a tutorial consisting of 12 weekly meetings in summer semester 2007 (each about 90 minutes). Such tutorial courses supplement one-term lectures about methods and techniques in HCI and requirements engineering, which are offered yearly to graduate students in our department. In most cases, students are familiar with programming, formal specification techniques and with software engineering methods but have few

experiences with a user-centred design process. Participation in the tutorial is optional. Students get no marks or points and their participation is not a prerequisite for other courses or examinations. Hence, the focus of a tutorial is typically on a continuous exploration of a single design problem which is guided but not predetermined by early design techniques.

In the example study, a group of six students was assigned to the task of analysing the software engineering (SE) course for undergraduates at our department. We were particularly interested in improvements for project work. The intrinsic interest of students in such a topic is discussed in [7]. The idea to describe “the life of a student’s project” emerged out of the analysis of two interviews we conducted with all members of a project group of the just finished SE course. To be more precise, with all members except F. who obviously caused much trouble in that group. One question we discussed was how an educator can provide better help to students to “transcend” possibly negative experiences with team work and to learn for further SE projects? The author (who was the tutor) suggested to create stories of student’s projects, which could be presented to the undergraduate students throughout a SE course.

The paper concentrates on describing how the story was developed. By doing so, it also shows the process that the students went through. Aspects of play and of HCI education become apparent. The description is based on material created and/or used during the tutorial course and on notes of the author. We used Trac¹ to store protocols and sketches created during meetings, to organize our notes about the story in a shared way (Wiki) and to access documents of a real student's project we used for our story. The tutor introduced design approaches such as scenario-based design to the students and provided guidance but apart from that she was a member of the group.

4. NARRATIVES IN HCI

The use of narratives has a long history in HCI. Personas are descriptions of users and their goals. The intention is to increase the understanding among designers about who use the system under design and why (e.g. [6]). Developers of use cases are recommended to write usage narratives to warm up before writing use cases as more abstract descriptions of how to apply a system [5].

Scenarios are used throughout the whole process in a scenario-based design. A scenario is seen as “a story about people carrying out an activity”, which in most cases is fictional but rooted in field studies to get knowledge about the problem domain, actors and their goals, plans and actions, the artifacts in use, but also about relations between stakeholders, their values and beliefs [11]. Rosson and Carroll describe scenarios as a “lightweight usage-centred design representation that keeps designers on the overall goal... a useful and usable system.” They suggest that scenarios help to involve all stakeholders into a vivid discussion about the new system and possible effects, e.g. on human activities. According to [2], an active participation of users is possible because scenarios can be constructed and explored in a shared way. (The actual writing may be done by single people though.) While software engineers tend to generalise situations and user actions, scenarios may help to illustrate the richness of activities because they are “rooted in specific situations from the domain

under scrutiny". They support the "tensions between reflection and action, between typical and critical situations and between plus and minus situations" [2].

Nielsen shows in [10] that users are mostly presented as flat characters in HCI scenarios. She argues that deep, believable characters should be developed instead. Plot-centred stories should be replaced by character-driven stories. Characters (with their goals and desires) have a number of traits and “a number of voices that interact with and against each other” and the “story develops because the character develops” [10].

An even broader view is taken in [12]. Wright and McCarthy point out the potential of novels to express the “open and continually changing and developing nature of experience and technology.” They refer to Bakhtin’s concept of a polyphonic novel as “the best tool for conceptualizing human experience that our culture has so far developed”. Again, the emphasis is on characters and the plot “becomes a way of creating scenes for intense dialogue with unforeseen outcomes” [12]. Wright and McCarthy criticise that not only many requirements engineering notations but also some user-centred representations tend “to present to users a finished image of themselves and their activities”. However, human experience is open. In a way, requirements are always incomplete. This should also be reflected in appropriate representations to encourage a true dialogue between users and designers.

5. THE IDEA DEVELOPS

5.1 First Sketches

Figure 1 shows one of two sketches created during the first discussion about the two project groups we had in mind to describe. In the end, we concentrated on the group in this figure. After seven more meetings the team members were elaborated.

Figure 1. First sketch of group I (names inserted later).

5.2 Plot

We identified ten scenes to illustrate the project work and the development of the characters in a coherent way.

1. first project meeting,
2. meeting: creation of the requirements definition,
3. meeting: discussion of the final version of the software requirements specification,
4. meeting: preparation of a first project presentation,
5. first project presentation.

¹ <http://trac.edgewall.org>

6. “after a long break”,
7. discussion between Thomas and Leo,
8. during the final coding phase,
9. Armin drives the others mad,
10. after the project: talk between Thomas and his older brother (who works as project manager).

Material for each scene was created or composed. Examples are given in Figures 2-4. One student who likes to write stories in his spare time wrote 1 to 2 pages narratives for two scenes. He plans to complete this work as part of a seminar paper. In Section 2, a translation of the beginning of the first scene was given.

5.3 Characters

From the beginning, Thomas was considered to be the central character (encircled in Figure 1). The notes in Figure 1 reveal that Armin is a direct reflection of F. (see interviews mentioned above) and others. He “talks much, does nothing” and likes to take the path of least resistance. More detailed, we imagined Leo as self-absorbed. He is a “Windows user”, spend his nights at the computer and sleeps during lectures. His initial attitude towards the project is somewhat lukewarm though he can be engaged if he is motivated. Gregor is convinced “Linux user”. He doesn’t like to feel dependent and is a bit of a loner. He wants to write good programs. He also wants to get good marks. Though three of the students are good programmers we wanted to depict the project as not successful. Thomas, the mediator in the group, was seen as highly motivated in the beginning and frustrated in the end.

Each story needs a premise the author(s) have to be convinced of [8]. One student proposed: “One grows from problems”. We later developed a more differentiated picture of Thomas. Though he seems to resign at the end of the project he may feel that managing a group does not mean trying to “control” everything, even with good intentions behind it. Scene 10 was added to give Thomas an opportunity for reflection. Gregor’s and Leo’s development may be more visible in the story. They learn to think more conceptually (e.g. scene 6). They also start to appreciate the use of models and representations and not only see them as something required by a teacher (e.g. scene 7).

5.4 Conflicts

The development of a character must be shown by conflicts they go through [8]. We needed to invent a concrete project for our story to be able to construct believable situations. There were three students in our group who have worked together in the undergraduate SE project. They remembered their project vividly and often referred to their experiences during our discussions. We decided to take their topic for our story: a bibliography management system which allows access to all references used in a working group and the composition of reference lists. But later, we also started to look into documents of their ‘real’ project to support our story. For example, we used parts of the requirements document for scene 2. In this scene, Thomas knows that he will be the only one who is prepared for the meeting. He hopes that the project progresses if he presents an almost complete document to the others. However, it turned out in the following meetings that the others didn’t internalize Thomas’ suggestions. The conversations often veer away from their purpose though Leo and Gregor start to try out things and also bring in good ideas.

Figure 2 shows material we composed for scene 7. Here, Leo and Thomas are sitting at lunch and begin to talk about the project.

Leo realizes that he needs to develop a more conceptual view on the life cycle of a publication. He feels a bit lost in his code. The friends begin to make drawings. The idea for this scene is rooted in real experiences. However, the right sketch in Figure 2 was developed during a discussion about the story.

Figure 2. Discussion about managing revisions (scene 7).

Armin is unreliable and a bad programmer. Gregor likes to solve problems but can be ignorant as well. For example, he doesn’t see the effort behind team formation. But then, he often needs to wait for Armin to be able to accomplish his own work and he starts to complain. When he sees the code Armin finally committed (illustrated in Figure 4), he “explodes”. The group couldn’t implement all required features in time. Although Armin is lazy and incapable indeed, he later may also serve as an excuse to the group for their “failure”. Figure 3 serves to illustrate the “coding behaviour” of the group in our story.

Figure 3. Final coding phase (scene 8).

17	private static int charToInt (char c){
18	int i;
19	i = 0;
20	if (c == 48){i = 0;}
21	if (c == 49){i = 1;}
22	if (c == 50){i = 2;}
23	if (c == 51){i = 3;}
24	if (c == 52){i = 4;}

Figure 4. Coding Horror (scene 9).

6. ANALYSIS OF THE EXAMPLE

6.1 Aspects of Play

Play is a closed, situated activity system which allows players to appropriate the environment in their own way by ignoring predefined meanings of artifacts and social relations. Not the artifacts and their mastery are in the focus but processes of self-perception and self-discovery. Yet, students were surprised that a story developed in our mind that they could identify with.

In the example, all students had experienced the work in an undergraduate SE project in former years. To create a story about such a project group was an opportunity to reflect own practices in a dialogue with the other members of the group. As “useless” as the discussions may have been perceived from the outside, they found their way into the story. Traits of characters had their originals. Situations were invented but were also a kind of “re-experience” of real situations as the students confirmed. We had the opportunity to look at project documents from a different than a software engineering perspective when we started to use pieces for scenes in our story. A fragment of bad code was suddenly seen as something “valuable” (for the story). Confusing diagrams or inconsistencies between different models or descriptions supported the story and were welcome. In the story, the project itself was described as a failure but the team members were not all depicted as “losers”. To play also means to assimilate negative experiences successfully.

6.2 Aspects of HCI Education

One of the tasks of HCI education is its integration into existing technology-oriented courses “to develop in students an appreciation for the importance of HCI issues in the overall acceptance and success of interactive software” [1]. HCI has contributed a lot to change our view on interaction design. In particular, the tension between abstract work descriptions as used in traditional engineering approaches and the richness of human activity and of actual use situations was emphasized. However, complementary design approaches like those mentioned in Section 4 are still often seen as something preventing the designers from doing their real work. Statements like “I can’t do anything with all the redundant information in a story”, are not a rarity among software engineers.

In this paper, we argue that play can help to deepen the appreciation of unfamiliar design perspectives and complementary approaches. In the example, students learned about narratives in HCI. However, more important was that they learned about themselves while playing with the idea to create a character-driven(!) story. This may have resulted in an acceptance of the “toys”. One further point deserves note. Egri’s book [8] is addressed to a general audience. In the introduction, he stated his hope that readers get a deeper understanding about the efforts of writing and develop more appreciation. Stories are engaging if the characters are no stereotypes but believable and if concrete situations are described. Readers get bored if a text contains details not necessary for the story. They also detect “inconsistencies” in a plot. It was surprising for some students to experience how hard it can be to develop “soft descriptions”.

7. CONCLUSIONS

Education and play are two different contexts of action which need to be interrelated in order to convey knowledge and skills to students but at the same time to enable them to explore unknown

paths. Engeström points out that development is often understood as vertical improvement of individuals (experts) though supported by social interaction and collaboration. However, development also needs a horizontal movement across ‘different worlds’ [9]. Such a movement includes the questioning of own assumptions and the understanding and appreciation of different perspectives.

The paper describes how technology-oriented students played with narrative approaches to reflect their own practices. It is shown that HCI education and play can support an intertwined vertical and horizontal development of students to prepare them for multidisciplinary work in interaction design. HCI offers many opportunities to relate education and play. One reason may be that still developing approaches encourage creative exploration. HCI teachers may be conscious of this situation and open to play.

In the illustrating example, students were not evaluated. They had no tight time frames. They were not forced to create a ‘product’ visible outside the group. There were few pre-defined roles and learning goals and no pre-established agenda. Such ‘playgrounds’ are necessary for playing with our education.

8. ACKNOWLEDGMENTS

My thanks go to Anja Heckt, Marcel Benthin, Stefan Piehler, Jonas Ritzke, Johannes Rössel, and Frieder Wartenberg, also for commenting on this paper.

9. REFERENCES

- [1] ACM SIGCHI Curricula for Human-Computer Interaction 1996. <http://www.sigchi.org/cdg/cdg3.html>
- [2] Bødker, S. 1999. Scenarios in User-Centered Design – setting the stage for reflection and action. In Proc. of HICSS’99.
- [3] Call for Paper: <http://hcie2009.abertay.ac.uk/cfp.html>
- [4] Carroll, J.M. 2003. HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science, Morgan Kaufmann.
- [5] Cockburn, A. 2001. Writing Effective Use Cases. Addison-Wesley.
- [6] Cooper, A. 1999. The Inmates are Running the Asylum. Macmillan Publishing Co., Inc.
- [7] Dittmar, A. and Forbrig, P. 2008. Some Thoughts about the Horizontal Development of Software Engineers. In Proc. Of EIS’08, LNCS 5247.
- [8] Egri, L. 1960. The Art of Dramatic Writing. N.Y., Simon and Schuster.
- [9] Engeström, Y. 1996. Development as Breaking Away and Opening Up: A Challenge to Vygotsky and Piaget. In: Swiss Journal of Psychology 55 (1996), 126-32.
- [10] Nielsen, L. 2002. From user to character – an investigation into user-description in scenarios. In Proc. of DIS2002.
- [11] Rosson, M.B. and Carroll, J.M. 2002. Usability Engineering – Scenario-Based Development of Human-Computer Interaction. Morgan Kaufmann Publishers.
- [12] Wright, P.C. and McCarthy, J.C. 2004. The value of the novel in designing for experience. In A. Pirhonen, C. Roast, P. Saariluoma, H. Isom (eds.). Future Interaction Design. Springer, 9–30.