Methodological Reflections on Working with Young Children

Matthias Korn
Aarhus University
Department of Computer Science
IT-huset, Åbogade 34
mkorn@cs.au.dk

ABSTRACT
This paper provides methodological reflections on an evolutionary and participatory software development process for designing interactive systems with children of very young age. The approach was put into practice for the design of a software environment for self-directed project management in intercultural computer clubs in Germany. Both, the process itself and insights gathered are described.

Keywords
Methodology, Participatory Design, Young Children, Software Development Process

INTRODUCTION
Doing participatory design with young children at the age of 6 to 12 years poses challenges that are different from those with teenagers or adults. Children in their early stages of cognitive development have difficulties in understanding complex and abstract concepts. They are much more sensitive to good evaluation design. Also, the cognitive development can vary greatly between children who are just one year apart or are even of the same age. On the other hand, they can be very refreshing and have a more creative and exploratory approach than teenagers or adults.

Druin and others already address many of the challenges and opportunities in her seminal book on designing technology with children [3]. This paper tries to enrich these with deeper insights gathered from a recent design activity with children in a computer club in a German elementary school. It especially highlights experiences with children of very young age.

The system being designed and evaluated (the process from which the following results are drawn from) helps children and other participants in the computer clubs to maintain an overview of their work and artifacts and to maintain a flow in their project-based collaborative activities. The system is proposed as a self-directed project management system. Early ideas can be found in [7]. The complete study of the system design and development is contained in [6].

First, the paper lays out context and setting of the study. Subsequently, the two phases of the development process, i.e. the pre-study and the actual co-development process are presented and relevant experiences with the both are discussed.

SETTING
The study takes place in two intercultural computer clubs for children and their parents (called *come_IN*) in Bonn and Siegen, Germany. The *come_IN* computer clubs are modeled after the Computer Cubhouses in the U.S. [8]. The concept has been adapted to the specific context in Germany. The focus of the project lies on integration of different cultures through shared practices and learning [10].

Since 2004 and 2006 respectively, the clubs in Bonn and Siegen meet for 2 hours each school week in the local elementary schools. Participants take part voluntarily and are usually students of the respective school. The children come from families of very diverse backgrounds: families of German decent or with migration backgrounds, from different economic and educational backgrounds. Some of the participants are relatively new to the club and others already participate for several months and up to two years.

Normal activities in the clubs are generally group-oriented project work and encompass the creation of personally meaningful artifacts in the form of photo stories,

Figure 1. An evaluation session with two participants.

animations, videos, games and other types of digital self-expression.

The study comprises 18 evaluation sessions of a software prototype being further developed during 9 club sessions over a period of two months in May and June 2009. The evaluations took place during regular club hours in the clubs' localities. First, single participants and later small groups of two or more participants were, together with a research assistant, involved in each evaluation session (see Figure 1). The participants were 13 elementary school children at the age of 7 to 10. Nine of them were male and four female. Seven out of the 13 participants have a migration background. Some of the participants took part in more than one evaluation session. Each evaluation session was documented as a screen cast with audio and video of the participants and video of the actual computer screen.

PRE-STUDY

A pre-study gathered first insights into the field and identified problems and peculiarities of the clubs. From these findings early requirements were derived and an initial, non-functional prototype proposed. This prototype is subsequently being evaluated and further developed in the next phase.

The pre-study is based on participatory action research [5] and encompasses participatory observation and note taking, informal as well as semi structured interviews and video analyses. Notes taken from memory supplemented the in-situ notes.

The pre-study took place in three consecutive club sessions in March and April 2008. Around 5 to 10 children and 1 to 3 parents were present at these sessions. At least 3 research assistants in each session took the role as tutors and observed the club activities and held interviews with participants.

The pre-study helped to get a better understanding of the field and identify first requirements, which were used as a starting point for the following phase.

EVOLUTIONARY, PARTICIPATORY S/W DEVELOPMENT

The non-functional prototype from the pre-study is now evaluated and further developed in an evolutionary (iterative development of individual functionalities or modules) and participatory (involvement of the children in each cycle) development process.

The software is deeply rooted in the everyday practices of the children. So it is crucial to include them and their views and opinions in the software development process. In terms of participatory design, the usage of the prototypes resembles a form of cooperative prototyping [1], where it is aimed to establish a shared understanding of the problem and solution space between designers and users through a co-learning process through iterative prototypes. The development process itself is derived from the spiral model [2] and the STEPS model [4], with the latter emphasizing the context in which the software is used.

Scaife and Rogers [9] give cause for concern that the imbalance of power between children and adults may have negative effects on the success of participatory design methods. On the other hand they report positive results when doing evaluation in pairs instead of one child alone, where two children encourage and build on the ideas of each other. A direct implementation of the design ideas of the children engaged them even more in the process.

GATHERED EXPERIENCE

Iteration of the prototypes proofed to be very fruitful. But it should be paid close attention that a meaningful sequential order is used to introduce the modules to the users. In this study, a login module was introduced before a registration module, which seemed logical to the author, as the registration would have been done only once per user and the login would be done each and every time they use the system. Still, although they had rarely any experience with software that had individual user accounts, children felt they first had to say who they are before they could choose themselves from a list.

According to cooperative prototyping breakdown situations should lead to in-situ modifications of the system with or with the help of the user. This is hardly possible within the provided context. Young children have a very short attention span and the busy surroundings of the clubs also hinder any concentrated efforts. A more low-tech and hands-on approach to prototyping (e.g. paper, scissors and other utensils to tinker) could be able to alleviate this problem.

In order to engage in free exploration of a new system, children need quite some context information on the idea and use of the system. They would not know what to make out of it otherwise. Additionally, many obvious opportunities to tinker with the system ought to be provided to make any exploration happen at all.

Scenarios were used to evaluate the software, which posed problems that children faced with the existing practices. Although it was often difficult for them to transfer the problems onto the prototype it generally helped to draw a comparison between existing practice and the new prototype. A group of two children had just experienced problems with finding their previously created artifacts. Quickly, the task was to similarly search for an artifact with the prototype. This way several strategies for searching became apparent which could be included in later versions of the prototype. Still, constructing meaningful scenarios was hindered by the fact that the early functionalities of the prototype were quite removed from the existing practices in the current work environment.

It is useful though to incorporate artifacts into the evaluation that have been created by the children themselves in previous sessions. This way they can better relate to the software and feel home. They are more confident in the evaluation situation.

At first, evaluation sessions were carried out with individual children. Together with the researcher they tried out the software prototype and gave moderate feedback on it. The evaluation situation often seemed a bit tense and directed by the researcher. In the second half of the study, the software was now evaluated in groups of two or three children, i.e. together with friends or siblings. This resulted in a more relaxed, playful and exploratory use of the software. The children incited each other and they became more engaged in the process.

It was of high importance that it has to be made clear to the children, that their opinion and feedback is highly valued. As equal design partners their ideas should be acknowledged and if possible directly implemented to see where this might go and that they are taken seriously. Children should be asked, "Can you help me?" instead of "Can I show you something?"

In summary, it is a challenge to integrate the evaluation of the software in the normal workflow of the children. The differences between the proposed software and the existing practices can be too far apart for the children to make a meaningful comparison of the two. The same applies to the scenarios as discussed above, which are difficult to create as the software in early iterations is not yet as powerful as the current work environment. Asking children directly about their opinion (e.g. problems or ideas for improvement) normally yields few results. Implications have to be drawn from observations and reading between the lines. The inferior visual design of the prototype and small bugs in the interaction did not seem to distract the children much from the actual evaluation. Also, drifting off in unimportant details could not be observed during evaluation. In regard to results, mainly problems pertaining usability of the software could be discovered fast and easily. In contrast to that, it was quite difficult to draw conclusions on the understanding of the underlying concepts and ideas behind the software from the children. They could only be interpreted very tentatively from the observations and conversations. The young age and their limited comprehension of abstract concepts as well as the distance to the existing practices are primarily made responsible for this.

CONCLUSION

Special methodological considerations have to be taken into account when designing software systems with children of young age in a participatory fashion.

Pairing children yields far better results than evaluating individually with one child. Consider low-tech prototyping in early phases or the development of new modules. Free exploration is not for free. Include many obvious opportunities to tinker with in your system. It should be absolutely clear what your system could be used for. Make clear and even better demonstrate that their opinion is valuable for you and that they are taken seriously.

ACKNOWLEDGMENTS

The work presented here was conducted as part of the author's Master's thesis at the group of Volker Wulf at the University of Siegen.

REFERENCES

1. Bødker, S. and Grønbæk, K. 1991. Cooperative prototyping: Users and designers in mutual activity. International Journal of Man-Machine Studies, 34(3): 453–478.
2. Boehm, B. W. 1988. A spiral model of software development and enhancement. Computer, 21(5): 61–72.
3. Druin, A. (ed.). 1998. The design of children's technology. San Francisco: Morgan Kaufmann Publishers.
4. Floyd, C., Reisin, F.-M., and Schmidt, G. 1989. STEPS to software development with users. In Proc. of ESEC 1989, University of Warwick, Coventry, UK, p. 48–64.
5. Kemmis, S. and McTaggart, R. 2005. Participatory action research: Communicative action and the public sphere. In N. K. Denzin & Y. S. Lincoln (eds.), Handbook of qualitative research (3rd edition., p. 559–604). Thousand Oaks, CA: Sage Publications.
6. Korn, M. 2009. Splatch 2: Projektmanagement für gruppenorientierte Projektarbeit in interkulturellen Computerclubs. Unpublished Master's thesis, University of Siegen.
7. Korn, M. and Veith, M. 2009. Learning support through scaffolding collaborative project work. In A. Dimitracopoulou, C. O'Malley, D. Suthers & P. Reimann (eds.), CSCL 2009 community events proceedings. Rhodes, Greece: ISLS, p. 73–75.
8. Resnick, M. and Rusk, N. 1996. The computer clubhouse: Preparing for life in a digital world. IBM Systems Journal, 35(3 & 4): 431–439.
9. Scaife, M. and Rogers, Y. 1998. Kids as informants: Telling us what we didn't know or confirming what we knew already? In A. Druin (ed.), The design of children's technology (p. 27–50). San Francisco: Morgan Kaufmann Publishers.
10. Stevens, G., Veith, M., and Wulf, V. 2005. Bridging among ethnic communities by cross-cultural communities of practice. In Proc. of C&T, Milano, Italy, p. 377–396.