Datalogisk Institut
KØBENHAVNS UNIVERSITET

DIKU

Proceedings of the Second Danish Human-Computer Interaction Research Symposium

7th November 2002

Edited by
Erik Frøkjær, Kasper Hornbæk
University of Copenhagen

Technical Report no. 2002/19
ISSN: 0107-8283

HCØ Tryk
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Classification: H.5

Preface

The Second Danish Human-Computer Interaction Research Symposium was held at the Department of Computing, University of Copenhagen, at the 7th of November 2002. The aim of the symposium was to stimulate interactions between HCI researchers through a mix of oral presentations, discussions of posters, and keynote presentations.

We received 17 contributions from 26 authors for the symposium which forms the bulk of these proceedings. In addition, the program for the symposium is included.

We wish to thank Microsoft Danmark for kindly sponsoring the symposium.

Københavns Universitet,
2002 October 16

Erik Frøkjær & Kasper Hornbæk

Contents

SECOND DANISH
HCI RESEARCH SYMPOSIUM

7th November 2002, Copenhagen University

INTRODUCTION

We are happy to invite researchers from academia and industry to contribute to the second Danish HCI research symposium. The aim of the symposium is to stimulate interactions between HCI researchers through a mix of oral presentations, discussions of posters, and keynote presentations. The focus on the symposium is on HCI research, e.g. usability work; novel interfaces; web design; affective computing; psychological models; computing in music, creative arts, design, and architecture; design of input devices; support of collaborative work; speech input; information navigation; mobile devices; learnware; visualization; home computing. However, reflections on and challenges of HCI work based on industrial experiences are highly welcomed as well.

PROGRAMME

9:30-10:15 Registration, coffee/tea, putting up posters.

10:15-11:00 Staffan Björk, "The Play Studio - Experimental Design in Ubiquitous Computing", PLAY research studio, Interactive Institute, Sweden.

> Staffan Björk is the research manager of the PLAY research studio: "The PLAY research studio investigates and invents the future of human-computer interaction. As computers become more and more a part of everyday life, the previous view of computers as strictly a work-oriented tool will change. We believe that in the future, computation will become just another material for design, and take a natural place in human existence alongside other basic technologies such as writing and electricity. The research in PLAY will prepare us for that future." http://www.playresearch.com/.

11:15-12:30 Oral presentations of papers:

- Anker Helms Jørgensen, "Computers: From Calculation to Culture – The HCI landscape in a Historical Perspective".
- Erik Frøkjær & Kasper Hornbæk, "Studying the Utility of Metaphors of Human Thinking in HCI".
- Marianne Graves Petersen, "Time and learning in HCI".

12:30-13:30 Lunch, small talk

13:30-14:00 Small talk, posters

14:00-15:00 Superflex, "Superresearch".

> Superflex are one of the most interesting and important artists' groups working today. There are three central members, Bjørnstjerne Christiansen, Jakob Fenger and Rasmus Nielsen, who are joined by various international collaborators on individual projects. Since 1995, they have worked on a series of discrete initiatives involving such issues as energy production in developing countries, internet television studios for specific neighbourhoods and communities and brand name copy production in South East Asia. Though very different, all these projects relate closely to questions of powerrelations, democracy and self-organisation. Superflex look at their works as

tools. The tools are considered as invitations rather than representations; invitations that call for a participation and continuation. The tools represent models that are being used by different persons or groups they are not "alternatives" but are continuations and show real behaviour patterns. http://www.superflex.dk/

15:00-16:00 Oral presentations of papers

- Jesper Kjeldskov & Mikael B. Skov, "Evaluating the Usability of Mobile Systems: Exploring Different Laboratory Approaches".
- Jakob Eg Larsen & Michael G. Rose, "Eight Fallacies of Distributed User Interfaces".
- Hans Dybkjær & Laila Dybkjær, "Experiences from a Danish Spoken Dialogue System".

16:00-17:00 Posters, coffee/tea

17:00-18:00 Oral presentations of papers

- Morten Hertzum, "Requirements Specifications and Scenarios: Two Design Artefacts in Software Engineering".
- Morten Borup Harning, "Structuring the User Interface Design Process - First Steps Towards a Frame-of-Reference for User Interface Design Issues".
- Olav W. Bertelsen & Søren Pold, "Towards the aesthetics of Human-computer interaction".

Posters/Demos will be given by:

- Tue Haste Andersen & Kenny Erleben, "Sound Interaction by use of Comparative Visual Displays"
- Olav W. Bertelsen, Toke Eskildsen, & Werner Sperschneider, "Comparative Visual Displays Use in the Oven – Product Development in HCI Research".
- Klaus Marius Hansen & Christian Heide Damm, "Combining Co-Located and Distributed Collaboration Tools".
- Jette Hyldegård, "Context based information behaviour and social interaction".
- Kristoffer Jensen, "Proposal for Verbal Attributes of Musical Sounds".
- Jannie Friis Kristensen & Christina Nielsen, "Interaction as Negotiation".
- Aran Lunzer, "Widget multiplexers for in-situ handling of alternative application states".
- Georg Strom, "Use of human centered stories for describing Human Computer Interaction".

PLACE

The August Krogh Institute, Universitetsparken 13, 1st floor, DK-2100 København Ø.

Several busses, e.g. 150s, go to Universitetsparken from Nørreport and Copenhagen central train station. See http://www.aki.ku.dk/aki/aki_details/gethere.html.

SPONSOR

The symposium is kindly sponsored by Microsoft Danmark, see http://www.microsoft.com/danmark/.

ORGANIZING COMMITTEE

Erik Frøkjær, DIKU - Department of Computing, University of Copenhagen, erikf@diku.dk & Kasper Hornbæk, Natural Sciences ICT Competence Centre, University of Copenhagen, khornbaek@nik.ku.dk

Sound Interaction by use of Comparative Visual Displays

Tue Haste Andersen
Department of Computer Science
Copenhagen University
Universitetsparken 1
DK-2100 Copenhagen
haste@diku.dk

Kenny Erleben
Department of Computer Science
Copenhagen University
Universitetsparken 1
DK-2100 Copenhagen
kenny@diku.dk

ABSTRACT

Navigation in pre-recorded sound for mixing with other sound sources is often done in contemporary music concerts. Commercial computer based DJ interfaces fall short when compared to their older analogue counterparts, the mixer and turntable. In this work we try to extend a computer based DJ system Mixxx, with visual displays of various sound parameters extracted from the sound. A demonstration of the system will be given.

1. INTRODUCTION

In contemporary music it is often necessary to synchronize pre-recorded audio with live instruments or other pre-recorded sound. This is a problem often carried out by a disc jockey (DJ) or a music programmer. The problem of matching different sources of audio in short time is demanding, and requires an interface supporting both fast skipping through large amount of sound, and at the same time fine grained control of position and playback speed.

This study focus on improving visualization in digital DJ solutions, based on parameters extracted from the audio, using DSP techniques. These include pitch analysis [8], transient detection [2, 4] and sound level estimation.

2. BACKGROUND

A traditional DJ setup includes two turntables and a mixer. This allows for controls such as continuous variable speed of playback, sound level, filters, and a cross fader for mixing between the two sound sources. Special CD players are also present in most setups. DJ CD players are most notably different from normal CD players in that they have speed control and abilities to store a specific position in a song. Even though the CD players have been available for a long time, and CD's are more convenient to transport when compared to vinyl records, the records still seems to be the preferred playback medium. Digital DJ software solutions are also available, making it possible to select and mix sounds stored in for instance MP3 format. These solutions often depend on the use of a mouse, and in some cases external knobs connected through MIDI. However, the problem is at best reduced to the same as for the DJ CD player. When comparing to the turntable, the DJ is loosing visual control by the loss of reflection from the grooves in the vinyl, and the ability to skip fast and precise through a song, by either spinning the record by hand, or by moving the stylus. The mixer allows for mixing of different audio sources into one, independent control of filters and sound effects, and a special output channel for headphones, which the DJ can use for listening and changing audio before it is mixed with the audio output. This is especially important in disciplines such as beat mixing, in which DJ's match two consecutive songs in beat before mixing them together, forming a smooth transition from one song to another.

3. METHOD

In the work presented here, we address the problem of giving visual feedback while searching through sound. This problem, is one of several earlier identified as part of the first authors ongoing research on interaction and visualization of sound.

To demonstrate the ideas in this project, and to perform usability experiments, an open source, cross-platform software mixer, Mixxx[1], has been developed. Mixxx provide features similar to commercial software for mixing pre-recorded sound. By using different interfaces to this software, comparative studies can be done, using qualitative, eg. think aloud experiments, and quantitative evaluation techniques.

For the quantitative studies the following hypotheses could be used when designing the experiments:

1. Visual cues improves the performers ability to find the "right spot" when performing beat mixing, and mixing with live instruments
2. Novice DJ's are faster at beat mixing using visual guiding displays (ex. parallel displays of beat information)
3. The time it takes to find a specific event, can be shown to follow the well known Fitts law. Applying Fitts law has previously been applied to the somewhat similar problems of pointing [1] and scrolling [5].

[1] See `http://mixxx.sourceforge.net/`

4. VISUALIZATION

Parameter visualization serves a number of purposes:

1. Provides cues of the structure of a song without the need to listen to the song, eg. by showing energy and tonality as function of time
2. Allows for matching of parameters from different audio sources using comparative displays
3. Supports collaborative work through overlay of real-world objects with visualizations of song parameters (Augmented Reality)

We plan to demonstrate a working prototype of a visualization module for Mixxx. The module will be based on OpenGL, to easily make use of the ARToolKit [6] or similar technology, at a later time. The parameters used are all time dependent: Waveform, average sound level, pitch/tonality and marking of beats.

These parameters will be drawn on a number of graphs for each song played back. The graphs show a large period of time, and by selecting a graph it will be displayed on a larger scale in the middle of the display. By simultaneous selection of the same parameter from different songs, the plotting of the values allows for visual comparison. In the prototype under development we wish to experiment with different display types [3], such as overview+detail and focus+context.

5. CONCLUSION

Ongoing work on improving navigation in pre-recorded audio has been presented. A prototype system is being built and will be demonstrated at the symposium. This prototype will later be used in evaluation of changes in usability, when compared to existing interfaces. A future perspective on the visualization of sound features, is direct manipulation of timbre characteristics, by use of timbre models [7].

6. ACKNOWLEDGMENTS

Thanks the people from the Music and HCI groups at DIKU, for feedback. The software implementation of Mixxx has been done by Tue and Ken Haste Andersen.

7. REFERENCES

[1] J. Accot and S. Zhai. More than dotting the i's - foundations for crossing-based interfaces. In *Proceeedings of CHI 2002*, April 2002.

[2] M. Basseville and I. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. 1993.

[3] S. Card, J. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization*. Morgan Kaufmann Publishers, 1999.

[4] L. Daudet. Transient modelling by pruned wavelet trees. In *Proceedings of the International Computer Music Conference*, 2001.

[5] K. Hinckley, E. Cutrell, S. Bathiche, and T. Muss. Quantitative analysis of scrolling techniques. In *Proceeedings of CHI 2002*, April 2002.

[6] H.Kato, M. Billinghurst, and I. P. et.al. Virtual object manipulation on a table-top ar environment. In *Proceedings of ISAR 2000*, October 2000.

[7] K. Jensen. *Timbre Models of Musical Sounds*. Ph.D. dissertation, Department of Computer Science, University of Copenhagen, 1999. Report no. 99/7.

[8] L. Rabiner. On the use of autocorrelation analysis for pitch detection. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-25:24–33, 1977.

Towards the aesthetics of Human-computer interaction

Olav W Bertelsen
University of Aarhus
Department of Computer Science
Aabogade 34, DK-8200 Århus N. Denmark
olavb@daimi.au.dk

Søren Pold
University of Aarhus
Department of Comparative Literature
Aabogade 34, DK-8200 Århus N. Denmark
pold@multimedia.au.dk

Abstract
We argue that the field of HCI needs a new aesthetic perspective to develop further, and to transcend current shortcomings.

Towards a new discipline
For years the ruling paradigm in human-computer interaction has been use oriented. From mechanistic approaches such as GOMS, over "user centred design", to participatory and radically process oriented approaches, minimising the intrusiveness of the interface in the situation of use has been the overall goal. Users should be able to do their work instead of dealing with the computer. The computer should be a tool, not the object of work..

Similarly in HCI, learnability has traditionally been looked upon in terms of how long it takes to master standard tasks (Shneidermann, 1990). More recent approaches based on activity theory have focused on how to support development in use, assuming that use basically cannot be anticipated at design time (Bertelsen & Bødker 2002). Whereas the traditional approach to learnability implies that "use" – what the application is used for – as well as the interface and the user are static entities that are being optimized against each other, the later approach implies a basic dynamics of the whole use situation, including the user, the task, the interface etc.

Designing for development in use means that design is extended into the process of use; the work arrangement of which the interface is part is continuously adapted to the evolving requirements of the supported practice. Even though the dialectic relation between tool and task is acknowledged, development is most often understood as aiming for smooth, transparent performance.

Experience oriented approaches that emerged in the mid 80's, in parallel with the use oriented, argued that interfaces should support the users enjoyable experience. Ironically, the Aristotelian poetics that e.g. Laurel (1986) built upon had the result that experience was reduced to the means for achieving efficient interaction. Aesthetics has largely been understood as the icing on the cake; the means for something else. Art should build the spectators character, invoke sensitivity to some important societal problem etc. In short, art and aesthetics has been said to have a purpose. With this "purposeful" (or intentional) aesthetics, experience orientation is at risk of becoming just a certain form of use orientedness (see e.g. Norman (2002) for a severe case of).

Concerns for transcending the well-known and well defined, has to a larger extend existed in the design literature. However somewhat subsumed under the tenet that new artefacts must fit the considered practice and that use should be brought into design through simulated work. In general approaches concerned with the cooperative design process have acknowledged a basic discontinuity between the old and the new design (Bertelsen & Bødker 2002). Thus, use-orientedness not only means a concern for technologies seamlessly fitting the working culture in question, but it also implies an emancipatory and transformatory program. But because design tends to be understood as the response to a state of need in some concrete practice innovation is often reduced to the mere adaptation of the technical work arrangement to changing conditions. Thus, the problem we are facing in HCI and design is the paradoxical one of meeting needs that don't yet exist, supporting the development of practice that we cannot yet imagine.

We suggest that considering the interface as a field of aesthetics can solve some of the current problems in HCI. To avoid sliding back into the dominating functionalism we further point to cultural and literary analysis as necessary elements in such a new discipline of HCI. The key to an instrumental understanding of the dynamics in the interface and the expansive potentials in design is aesthetics.

Questions of aesthetics are not new within HCI, but until recently the aesthetics has mainly been subordinated under functionalism as the icing on the cake. But computers and interfaces are not limited to workspaces and use situations where functionalism and effectiveness is key. With digital art, the Internet and computer games, what have been termed as cultural interfaces (Manovich, 2001) are flourishing -- interfaces that are not transparent or functional but evident, quixotic, and highly visible. In fact most cultural interfaces (within art, computer games, web design) are aiming at originality instead of standards and guidelines -- some of them are, like modern art, searching for original ways to express themselves and their mediated nature. Instead of the standardized transparent interface, these interfaces aim for entertainment, experience and self-expression. Besides, functionalism has in HCI as well as elsewhere (architecture, urban planning) demonstrated its shortcomings in both dull interfaces and/or the function-obsessed interfaces of e.g. modern office software and operating systems.

Viewed from the perspective of the history of aesthetics, the interface is in the process of developing a formal language of its own. From being seen merely as a transparent tool emulating something else (an office tool, a control board of a machine, a traditional media technology, etc.), the interface gets increasingly visible as a phenomenon in our culture and as a formal, aesthetic category developing its own expression, art and culture. Just like photography before it, the interface becomes an aesthetic form in itself, and in the process changing both the cultural potential and perception of itself, but also of older media and aesthetic forms. In fact, as argued by dialectic materialists (Benjamin, 1974; Wartofsky, 1973) and various media theorists (McLuhan, 1994, Debray, 1994) it is the very constituents of our sense perception that is undergoing changes when our media change. The interface is, as media technologies before it, changing the range and scale of our sense perception; with the interface information becomes visible, interactive and thus is increasingly becoming an important part of our perceived reality.

We aim to work with a materialist perspective in order to sketch out new theoretical and practical perspectives for HCI considered as an aesthetic discipline. Our current slogan is:

Computers and interfaces are real, not virtual, and computer applications have effects on reality, on how it is perceived, and how it is constructed.

We must leave the utopian, 'virtual' thinking so common around computers and learn from the materialism found in literary and art historical realisms in order to put interfaces in material and cultural context. This could mean interfaces that are provocative, user-unfriendly (Dunne, 1999), humorous instead of aiming at being transparent, universal etc. Human-computer interaction is a difficult translation, a dialectics -- sometimes even a dichotomy -- between human and machine, and this translation should not always appear automatic, smooth, and seemless. Instead some of the underlying structures of the software and machine should be displayed, just like good works of art display and contain material and representational self-reflection. This could lead to critical interfaces, that gives the user insight into to the workings of the machine and software, which would also give the user better possibilities to develop unforeseen and 'un-designed' uses. Perhaps a deconstructive interface design under the slogan "What You See Is What It Does". A new design praxis aiming at honest interfaces which deliberately and evidently negotiate between the logic of the computer, the intended and designed interaction and the possibility of new, unforeseen interpretations of the user.

First step in our work is detecting and discussing whether this is already taking place in experimental interfaces, computer games, digital art, etc. In this sense, we intend to discuss interface design that is culturally, perhaps subculturally, coded instead of universal and user friendly; seducing instead of transparent. Interfaces with atmosphere and style, perhaps even other styles than the renaissance windows (try with some baroque opacity or surrealistic inscrutability), other ways of staging the interaction between human and computer (what about staging the computer as something other than servile?), alternative figurations (why has HCI only discovered metaphor and not metonymy or allegory?), critical interfaces, essayistic interfaces. In short interfaces, where the aesthetics is on equal level as the functionality and not subordinated to it. As ways of designing we might look at how artists and artistic designers work with the materiality.

HCI needs aesthetic perspectives. Recent developments within HCI have explored such directions (e.g. RCA, Play Research, ID Studiolab). Still there are lots of problems with how to introduce aesthetics in HCI and what the outcome of it could and should be. It is obviously, that computer games, artistic and entertainment-oriented interfaces need aesthetic approaches, and that HCI cannot deal sufficiently with these kinds of interfaces today. However, work-related interfaces need aesthetics too. We continually run into conceptual and practical problems suggesting that the balance between functionality and aesthetics is very uneasy indeed. We do not see any easy an fast solutions, beyond establishing a long committed joint discourse involving HCI, literature and other aesthetic disciplines.

References

Benjamin, W. (1974-1989). Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit. In *Gesammelte Schriften,* Frankfurt/M: Suhrkamp,., bd. I,2, 471-509

Bertelsen, O. W. & Bødker, S. (2002). Discontinuities. In Floyd, C., Y. Dittrich, R. Klischewski (eds.) *Social thinking - Software Practice*. MIT press, Cambridge, MA.

Debray, R. (1994). *Manifestes médialogiques*. Gallimard.

Dunne, T. (1999). *Hertzian Tales - Electronic products, aesthetic experience and critical design*. London: Royal College of Art.

Laurel B. A. (1986): Interface as Mimesis. In Norman, D. A. & Draper, S (eds.) *User Centered Systems Design*. Erlbaum.

Manovich, L. (2001). *The Language of New Media*. Cambridge, MA: The MIT Press

McLuhan, M. (1994). *Understanding Media – The Extensions of Man*. London: Rout-ledge.

Norman, D. A. (2002). Emotion & design: attractive things work better. In *Interactions 9,4*: 36-42. NY: ACM Press.

Shneidermann, B. (1990). *Designing The User Interface 2nd edition*.

Wartofsky, M. W. (1973). Perception, representation, and the forms of action: toward an historical epistemology. In Wartofsky, M. W., *Models*. Dordrecht: D. Reidel 1 Publishing Company, 1979. pp. 188-210.

Use in the Oven – Product Development in HCI Research

Olav W. Bertelsen, Toke Eskildsen, Werner Sperschneider

Centre for Pervasive Computing, University of Aarhus
IT-Parken, Åbogade 34, DK-8200 Aarhus N
{olavb|darkwing|werners}@daimi.au.dk

ABSTRACT

We report on a collaboration project between a small manufacturer of ovens and a group of researchers. The project ensured, through field studies and prototyping, that a new generation of the products were usable, but it was also a learning experience for the manufacturer. The project is a fruitful model for future collaboration between industry and academia.

INTRODUCTION

Over the last year we have been engaged in a development and research project together with Hounö A/S, a local manufacturer of ovens. The cooperation was initiated by the manufacturer, motivated by the fact that their existing line of interfaces suffered from being overcrowded with functions added over the years, and not having a modern appearance (fig. 1). Previous efforts to redesign the interface had not been successful; therefore they decided to team up with an external partner to complete the redesign.

The overall design dilemma was to make an interface that supports programmability and other advanced features, and at the same time looks like an oven. Thus, not alienating the majority of the users who were not at all confident with the idea of using a computer-like device in the kitchen.

Fig 1: The current interface

Our motivation, as researchers, was that the project provided a real-world context for studying (fairly uncomplicated) pervasive computing technology in use, and that it provided an opportunity to commit our knowledge production closely to design. Right from the beginning, project time lines were tight, and several basic decisions regarding the new interface had already been made. Most importantly, the use of a colour LCD display and two rows of soft keys in an ATM like layout. This limited the design space, but at the same time made it more likely for the manufacturer to have a new product ready on time. We saw these limitations as an interesting challenge.

PROJECT ACTIVITIES

Among the different things we did in the project, two types of activities proved to be particularly fruitful. Firstly, we conducted field studies of the use of the present production model line as well as the use of competitor products. Secondly, we developed a prototype reflecting the changing design specification, and we tested it with users both in the lab, and on site.

Disregarding the fact that a new design had already been developed our first activity was a general study of ovens in use. The aim of this was to find out what kind of device an oven is, to transcend the narrow idea of the oven as an assemblage of functionality. We conducted observations in three types of settings defined by the manufacturers product line: the bake-off oven for bread baking in small shops, the combi steamer oven for institutional kitchens like hospitals, and ovens for À la Carte restaurants and catering services. As part of the development project, studying ovens in use was important as a way to establish a pool of reference points in the further construction process

The other main activity was the continuous construction of a prototype reflecting the evolving state of the manufacturers specification. Normally in a user centred design project, we would have based the first prototype on insights from field studies and other interaction with users. However, in this project we started out by a straight implementation of a specification developed by experienced electrical engineers with the manufacturer in cooperation with a graphics designer. In parallel, various design drafts were produced aiming to highlight and solve the major issues found in the field studies. The first versions of the prototype were operated with a mouse on an ordinary desktop computer. In the next round we build a mock-up of an oven front with real keys, and a laptop computer hidden behind the front only showing the part of the screen corresponding to the area of the LCD screen planned for in the new oven. This mock-up was brought into the field and tested with ordinary users in their own context (fig 2).

Through prototype testing in the field more issues were identified. In short an iterative process took form in which the initial field studies, the original design specification, the insights from implementing the

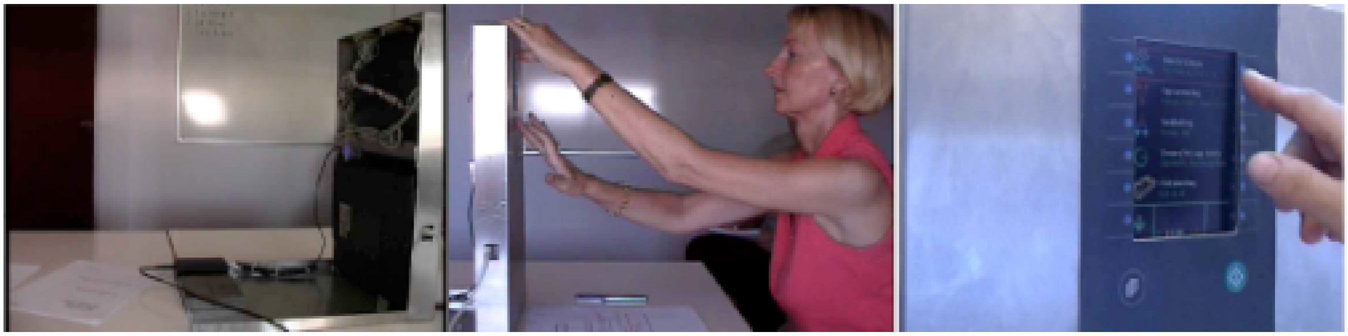


Fig 2: The prototype

specification and the field test results were interacting in the refinement of the evolving interface.

The current status of product development is that the oven will go into production in the first half of 2003. The interface still needs a few more iterations, but everybody in the project group and in the company are confident that the new interface is eventually to be brought to market.

DISCUSSION

As our cooperation with the oven manufacturer is both a development project and a research activity, we have to discuss it in these two distinct perspectives.

The project as a development activity

The manufacturer gained insights into the potential use of the new design. This enabled them to judge the new design based on a realistic basis rather than opinion. Thus, radically reducing the risk of introducing a completely new interface on their products.

The field studies became an efficient infusion of concrete use into the design loop. In several situations "design by pure reason and long experience" was corrected by observations of what people actually did with their oven. The most prominent example of this was the way we solved the seemingly un-solvable issue of cramming more information onto the display than there seemed to be room for. By referring to observations of use and the testing of earlier versions of the prototype it was possible to make decisions that from the point of view of "pure reason" would be illogical and inconsistent, but from the point of view of actual use proved to be consistent.

Building the prototype proved, not surprisingly to be an effective reality check, bringing the design team away from mere speculation and logical reasoning. We experienced over and over again that building an idea into the prototype yields much more basis for assessing it than we could get from reviewing a careful specification of it.

To a large extend the project became a learning experience for the manufacturer. They learned that users' reactions to a new product are fundamentally impossible to predict, and that engineers therefore should not design alone, but also take time to involve real users as a resource. They learned that it is possible to do testing in context and get useful and representative information, and they learned enough practical tricks to start doing testing themselves. They experienced that testing in the field could reveal surprising features of the interface, things they believed to be easy to understand proved to be very hard for the users to grasp and aspects they expected users to have a hard time dealing with appeared to be easy and straightforward for the users. Finally, they learned that instead of patching new functionality onto the interface, technically meaningful extensions should be backed up by ongoing studies of the ovens in use, possibly engaging sales and customer support actively in maintaining continued contact with costumers as use evolves.

The project as a research activity

From the point of view of research the project is far from finished; we are just now about to begin the analysis of the oven in use. This analysis will draw on both the initial field studies and the prototype testing. For this analysis, the insights into the specifics of the technical substance of an oven, has already proved to be valuable. We might have been able to make a field study of ovens in use without cooperating with the manufacturer, but we would not as easily have been able to get an understanding of the technical substance.

These insights should be weighted against the resources spend on developing the evolving prototype. However, from the point of view of the research project that has been almost free in the sense that it was build as part of a master thesis project and partly paid by the manufacturer. Apart from the concrete implementation, we believe that the effort put into the development activities consistently have contributed to the construction of an understanding of the oven as mundane pervasive computing.

We believe that the project is a good model for future cooperation with small manufacturers of equipment with an increasing amount of computing power. And we hope to be able to use the project as a pilot experiment for a much larger engagement with and handful of manufactures of different types of equipment.

Acknowledgements

First of all thanks to Hans Harbeck and other folks at Hounö A/S. Susanne Bødker and Christina Nielsen have provided valuable comments on earlier versions of this paper. The project has been funded through The Centre for Pervasive Computing, contract #222.

Experiences from a Danish Spoken Dialogue System

Hans Dybkjær
Prolog Development Center A/S (PDC)
H. J. Holst Vej 3C-5C
DK-2605 Brøndby
www.pdc.dk/speechlogic
+45 3636 0037
dybkjaer@pdc.dk

Laila Dybkjær
NISLab, University of Southern Denmark
Forskerparken 10
DK-5230 Odense M
www.nis.sdu.dk
+45 6550 3553
laila@nis.sdu.dk

ABSTRACT
Industrial speech recognition for the Danish language has been available for the last 2-3 years. We describe experiences from one of the first and so far most complex dialogue systems in Danish, aimed at providing general information about holiday allowance.

Categories and Subject Descriptors
H5.2 [**Information interfaces and presentation**]: User Interfaces - *Voice I/O*. D2.2 [**Software engineering**]: Design Tools and Techniques - *user interfaces*. H1.2 [**Models and Principles**]: User/Machine Systems - *Human factors*.

General Terms
Design, Experimentation, Human Factors, Languages.

Keywords
Spoken dialogue, industrial experience.

1. INTRODUCTION
For the last 2-3 years industrial level speech recognition for task specific natural language dialogue has been available for Danish. In year 2000 a market investigation of recognisers showed that there was one Danish recogniser, produced by Philips, for over-the-phone speech recognition. In autumn 2001 a Danish recogniser from Nuance appeared, also for over-the-phone speech, and now IBM plus NST seemingly are on their way. Philips also offers open microphone recognition, typically dictation systems specialised to a particular area.

We are aware of one such Danish dictation system which is meant for doctors. As regards spoken over-the-phone applications in Danish there are still only a few of them, including a country code information system, a flight timetable and arrival information system, an address information system, and the holiday allowance information system presented in this paper (+45 4820 4910 code 3).

A general interest in the area is emerging. But it is at the same time clear that there is quite some work to be done to cultivate a real market. The experience from other countries where spoken dialogue systems have been available for several years (the first commercial system appeared in the US in 1989) is that such systems can certainly save money for the company and at the same time increase the service to customers. But there are many issues to be aware of when making a commercial dialogue application. Some of these dialogue application issues are discussed in books, such as [6]. We will address some of the issues we have experienced while developing one of the first and at the same time most advanced commercial Danish dialogue systems.

Second Danish HCI Research Symposium 2002, November 7, 2002, Copenhagen, Denmark.

2. THE HOLIDAY ALLOWANCE PROJECT
The above-mentioned market investigation also looked at the market interest in spoken dialogue systems in Denmark. Information was collected via a questionnaire sent to about 200 companies and institutions. One of the very interested respondents was ATP (The Danish Labour Market Supplementary Pension Scheme). Together NISLab and ATP applied for money to initiate development of a spoken dialogue system. They obtained funding from The National Agency for Enterprise and Housing. After a call for tender the Danish software house PDC was chosen as software developer and the project started in autumn 2001. The agreed upon system was in the area of holiday allowance (Danish: feriepenge) which is administered by FerieKonto at ATP [3]. For all employees 12,5% of their salary is saved each month as holiday allowance. Employees continuing in a non-temporary position will just get their ordinary salary but if they change to another position or have a temporary position, they will get a holiday allowance certificate and will get allowance during their holiday by filling in this form and submitting it to FerieKonto. The rules concerning holiday allowance are quite complicated so many people have a need for information.

So far FerieKonto has via a voice-response system offered very general information on what to do if you have a holiday allowance form and on when you will get your allowance. In addition FerieKonto has offered access to a web page with answers to frequently asked questions.

ATP constantly aims at improving their service to customers while at the same time keeping the costs at a minimum or even reducing them. They also have as a goal to be on the front edge of development. Taken together this was the motivation for ATP for being so interested in a spoken dialogue project. They know that there is still a considerable number of their customers who either don't have access to the internet at home or who don't use the web pages even if they have access. On the other hand, nearly all people have a phone and know how to use it. However, making answers to frequently asked questions available over the phone requires much more than an ordinary voice-response system.

3. TWO-STEP APPROACH

It was agreed to take a two-step approach to the holiday allowance system to be developed by PDC and NISLab. The first part of the system would be a fairly simple general information system which would only have a slightly larger coverage than the existing voice-response system for general information. We have called this first part Vejled (Guidance). The second step would enhance the first part into a Frequently Asked Questions system which we have named FAQ. The primary goals of Vejled were (i) to get the technology into place while still having a relatively simple dialogue system, and (ii) to generate initial experience with real users. We knew that the FAQ system would be a very challenging system to address with its unstructured task which is difficult to handle in a reasonable way without adding too many annoying dialogue constraints.

4. USERS AND DATA

The end-users for the system are FerieKonto customers, i.e. any person who has holiday allowance and has questions about the holiday allowance rules. The system is a walk-up-and-use system, which means that it is intended for users who have no prior knowledge of the system. Interaction is via speech only over the phone.

Another type of user is the system administrator at FerieKonto. His interface to the system is different from that of the end-users. It requires some training to learn how to administer the system, and interaction is via a normal GUI and not via speech. In the following we only consider the end-users.

We wanted to involve users from early on in the development process. As mentioned, in principle any employee is a potential user of the system. However, it is not without problems just to use any such user. The main problem is that they only have a fictive need for information expressed in the scenarios given to them in writing or orally. Also such users know they are part of a test and tend to be more patient than one would be in a real situation. For Vejled we started by using colleagues as test users. They were typically just briefly informed about the system and asked to call. We made a first few rapid iterations in this way. All calls were transcribed and analysed resulting in changes to the system. The amounts of data were small since people didn't phone more than once unless explicitly told to do so. During spring 2002 we invited people outside our sites to call the Vejled system. This resulted in 225 calls which were transcribed. Transactions in the dialogues were carefully analysed. By a transaction we understand a piece of information the user tries to get from the system. In one dialogue the user may make several transactions. A coding scheme was developed for the mark-up of transactions and a coding tool developed by PDC was used for the actual annotation. This coding tool includes statistics so that e.g. transaction success could be easily calculated. The transaction success was 91.8 for these calls.

People were encouraged to fill in a questionnaire but despite the 225 calls we only received 12 filled questionnaires. On the average users were quite positive but clearly there was still room for improvements.

We have also made closely monitored lab-tests. They require quite some effort and don't provide a lot of language data but per call they certainly provide by far the most information on which improvements to make.

Real data can only be collected with real users and that is what we are doing for Vejled now as reported in Section 5. For FAQ we are still running lab-tests and tests involving subjects who volunteer to call the system, but we expect to have collected more experience with this system before the symposium takes place.

5. OPERATION

At the time of writing Vejled has been in public operation for more than a month. Since Vejled has been put in operation we have collected almost 1000 calls to the system by the public which are pretty many calls taking into account that we are outside the holiday season. All calls have been transcribed and are now being analysed. We expect to report on the findings in these dialogues at the symposium.

6. TECHNOLOGY

Both Vejled and FAQ are implemented on the Philips SpeechMania platform [5], which includes a recogniser, language processing, and tools for output generation and transcription. In order to build a spoken dialogue system on this platform one has to implement the dialogue manager, add lexicons and grammars for the task in question and record output phrases unless one decides to use speech synthesis. Considering price and customer quality, recorded phrases were chosen for both Vejled and FAQ. However, the system is prepared for synthesis as parameters of synthesis continuously improve. In terms of development and maintenance and for dynamic systems synthesis is much to prefer.

The implementation is done in HDDL (higher dialogue definition language) which is an event-driven imperative language for dialogue management with a C-like syntax, supplied with declarative grammar and knowledge rules for modelling language and semantics.

The implementation employs standard hardware such as IBM Windows 2000 servers and NMS telephony boards.

7. DIALOGUE DESIGN

Two example dialogues are given in Table 1 at the end of this paper. Some of the design issues we have had to address include:

Introduction: The introduction must indicate to the users what they can expect from the system, and what they can say. Also, the introduction must be inviting. Many users hang up immediately during the introduction phrase, without trying to speak to system. The reader is encouraged to consider how the introduction in the Vejled example in Table 1 meets the requirements of being inviting, telling what can be done, how to do it, and not being too long, at the same time.

The "rhythm" of the dialogue: A natural dialogue must take into account issues such as the length of information blocks, when to inform and when to ask questions, detection of back-channelling, when to make pauses and how long silences to accept.

Help: First of all we need to detect that there is a problem. If a problem is detected, we need contextual instructions which offer more elaborate help than some general help function would do, and ultimately personal assistance should be offered.

Meta: In general, the ability to continue smoothly in case of miscommunication, and to make clarifications must be available. There are several examples in the FAQ dialogue below.

Language: Issues include vocabulary, syntax, and statistical evidence. The latter is very important in today's recognisers and requires large amounts of data. For speaker independent telephony applications transcription of 2-10 hours of user input from application realistic dialogues is needed. For dictation systems, 100-500 million words of text of the type to be dictated are required. The system should accept input as close to natural language as possible within the given domain, so transcribed material from user tests is also used to improve the system's coverage.

Feedback: The system must make clear to the user what it understood and what it is doing and why. The feedback may be explicit (query for confirmation), implicit (say what has been understood, but continue without waiting for confirmation) or implied (the very act of the system shows what has been understood). The FAQ example in Table 1 shows examples of all three cases.

Prompts: The system's vocabulary, formulations, voice and other effects such as music greatly influence both the user's perception of the system, the user's language, and the user's understanding.

Information management: The domain information, such as the holiday allowance rules, addresses, and opening hours, is quite huge and will even change over time. An issue is how to specify the dialogue so that it can be communicated to and understood by the customers and their domain experts, i.e. in our case ATP and people at FerieKonto, and so that it – once it is implemented - can be easily maintained. In the reported system we have defined a domain specific representation in XML, which is then compiled into the executable HDDL (see Section 6) code.

8. RESEARCH AND ENGINEERING

The basic recognition technology has been available for more than 10 years although considerable improvements have been made up through the nineties. Specifically for the Danish language the appearance of Danish recognisers within the last couple of years has been crucial to the step from research systems only to commercial applications.

In our experience the co-existence of research and engineering can be very fruitful. Often, research is ahead of technology. The basis of today's dialogue design was refined by research in the mid-nineties [1], but e.g. barge-in as well as the use of recognition scores for controlling feedback [2] have just recently become technologically reliable. Recent research demonstrates that prosody can be used for detecting corrections and aware sites [4], but today's recognisers do not support this feature. Here engineers owe a lot to basic research.

It also happens that technology is ahead of research. For example, input provides clues, such as references, negation and affection, but we do not know how to handle these in an operationally tractable and robust way. The problems have wider perspectives for our general knowledge, and the concrete problems observed in industrial applications may serve as motivation for research.

Finally, there is much more to making a customer application than what is addressed by research:

Telephony technology: Every new combination of PBX (switchboards), telephony boards, and IVR/speech platforms is almost certain to yield new, hard problems. Related is also dimensioning: How many lines, how many licenses, how many and powerful machines?

Real world size problems: The FAQ specification is some 50 pages long, and compiled into HDDL more than 15.000 lines of high-level code. This is a lot of data to explain to customers and co-developers and to administer during maintenance, and means of presentation and representation must be designed.

Testing: Does the application run robustly, meeting requirements of transaction success rates and runtime with no breakdowns?

Documentation: The system might have to be taken over by other people; legally it is not developed by single persons but by companies.

Administration and maintenance procedures: Both data handling, annotation tools and standards, installation and update procedures need to be specified and developed. Decisions must be made on work division between the software vendor and the customer's system administrators, and the latter must be trained.

Marketing and sales: Considerations of how to sell more are crucial.

Finalising: A research project can be a success even if it just ends giving some good experiences. An industrial project must end with a product that satisfies the customer.

9. ACKNOWLEDGEMENTS

The project is supported by The National Agency for Enterprise and Housing, The Danish Ministry of Economic and Business Affairs. We would also like to thank the staff at ATP for kind, competent and enthusiastic cooperation.

10. REFERENCES

[1] Niels Ole Bernsen, Hans Dybkjær, Laila Dybkjær: *Designing interactive speech systems*, Springer 1998.

[2] Hans Dybkjær, Niels Ole Bernsen, Laila Dybkjær: *Inferential issues in a spoken language dialogue system*, Proceedings of FQAS'96, Roskilde University, Denmark, 1996.

[3] FerieKonto/ATP, www.FerieKonto.dk, 2002.

[4] Julia Hirschberg, Marc Swerts and Diane Litman: *Labeling Corrections and Aware Sites in Spoken Dialogue Systems*, Proceedings of the Second SIGdial Workshop on Discourse and Dialogue, Aalborg, Denmark, 2001.

[5] SpeechMania, www.speech.philips.com, 2002.

[6] Susan Weinschenk and Dean T. Barker: *Designing Effective Speech Interfaces*, Wiley, 2000.

Table 1. Two example dialogues - translated from the Danish

Vejled: Complete customer dialogue from the Vejled system.	**FAQ:** Subset of user test dialogue from the FAQ system.
Very typically the customer calls, selects his/her situation, gets the information, and exits immediately. If the user does not hang up, the system will continue to offer operationally related information such as address and email-address, and to start all over again. Also not shown is the pop up of contextual instruction and help prompts whenever problems occur.	This dialogue typically gets much more conversational, with topical changes and meta-communication. Note e.g. the extra confirmation in S 3.1, the prompting S 6.1 when the user says nothing, and U 7.1 where the user clarifies that s/he meant transfer of holiday. Also note the use of barge-in in U 7.1, a feature also present in Vejled.
S 0.1 - Welcome to FerieKonto's speech based guidance system S 0.2 - Here you can get standard information about payment of holiday allowance. - You may always ask for help or repetition. S 0.3 - Say if you - are an employee or unemployed in Denmark - are unregistered with the national register and live abroad - or have left the labour market due to age or health. U 0.1 I am unregistered with the nat- the national register *Recognised I am unregistered got the national register* S 1.1 - Unregistered with the national register. S 1.2 - Do you live in Denmark? U 1.1 no *Recognised no* S 2.1 - If you are unregistered with the Danish national register and do not work in Denmark S 2.2 - You can get your holiday allowance transferred to a Danish or foreign bank. - You must send your bank account number by e-mail, fax or mail. - Remember to write your civil registration number. - For foreign banks you must also write the detailed address of the bank. - A transfer to a foreign bank costs 75 DKK in exchange fee. *Event 1 nothing recorded* S 3.1 - If the FerieKonto certificate is correctly filled in when we receive it, we transfer the money to your account. - We need about five working days. - You cannot get your money until one month before the first day of your holiday. *Event 2 hangup*	? ? ? S 2.1 Ask about how you earn holiday allowance, get a FerieKonto certificate, fill in your certificate, send in your certificate, or get your money. U 2.1 how do I earn holiday allowance *Concept contribution(how do I earn holiday allowance)* S 3.1 - You asked about - earning S 3.2 - Your employer must set aside 12,5% on top of your salary. S 4.1 - Some employers have an approved holiday allowance scheme and thus they need not pay to us. - In that case you must contact the employer directly to get a holiday allowance certificate. S 5.1 - You may also ask about problems with your employer's payment of holiday allowance or about transfer of holiday to the next holiday year. *Event 4 nothing recorded* S 6.1 - Sorry. - What did you say? U 6.1 I would like to know something about transfer of holiday allowance to the next holiday year *Concept accept(would like to)* *Concept next(next)* *Concept money(holiday allowance)* *Concept transfer(transfer)* S 7.1 - The money is transferred to the account that appears from your FerieKon- U 7.1 @ transfer holiday *Concept reject(nope)* *Concept holiday(holiday)* *Concept transfer(transfer)* S 8.1 - Usually you must take your holiday before the end of the holiday year. - However, your union may have an agreement that allows for transfer of holiday. - You may then arrange with your employer that holiday beyond 20 days is transferred to the next holiday year. - Otherwise we transfer your undrawn holiday allowance to the Labour Market's Holiday Fund which will use the money for collective holiday purposes. ? ? ?

Studying the Utility of Metaphors of Human Thinking in HCI

Erik Frøkjær & Kasper Hornbæk
Department of Computing & Natural Sciences ICT Competence Center
University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark
+45 35321456 & +45 35320393
erikf@diku.dk & khornbaek@nik.ku.dk

ABSTRACT

Understanding human thinking is crucial in the design and evaluation of human-computer interaction. Inspired by introspective psychology, we present five metaphors of human thinking. The aim of the metaphors is to help designers to consider important traits of human thinking when designing. The metaphors capture aspects of human thinking virtually absent in recent years of HCI literature. As an example of the utility of the metaphors, we have shown how a selection of good and poor user interfaces can be appreciated in terms of the metaphors. The metaphors can also be used to reinterpret central notions in human-computer interaction, such as consistency and information scent, in terms of human thinking. Finally, we have experimented with using the metaphors of human thinking for usability inspection. The proposed inspection technique makes users' thinking the centre of evaluation and is readily applicable to new devices and non-traditional use con-texts. Initial experience with the technique suggests that it in discussing and evaluating user interfaces is quite effective.

1. INTRODUCTION

For some years our research and teaching in human-computer interaction have been inspired by William James's and Peter Naur's descriptions of human thinking [7-11]. Similar descriptions along with many brilliant design discussions have lately been introduced to HCI in Jef Raskin's book 'The Humane Interface' [13]. Naur's and Raskin's work are complementary to most psychology used in HCI, but is supported by extensive evidence from classic introspective psychology [7], and from experimental psychology and neurology [1,2]. Several of the aspects of human thinking described in this work are of critical importance to human-computer interaction: (1) the role of habit in most of our thought activity and behaviour—physical habits, automaticity, all linguistic activity, habits of reasoning; (2) the human experience of a stream of thought—the continuity of our thinking, the richness and wholeness of a person's mental objects, the dynamics of thought; (3) our awareness—shaped through a focus of attention, the fringes of mental objects, association, and reasoning; (4) the incompleteness of utterances in relation to the thinking underlying them and the ephemeral nature of those utterances; and (5) knowing—human knowing is always under construction and incomplete.

2. METAPHORS OF HUMAN THINKING

Below metaphors of human thinking (MOT) is summarized by describing the underlying understanding of human thinking and the five supporting metaphors. We also give examples of how to use the metaphor to conveniently understand good and poor interfaces, and outline key questions to consider in a usability inspection. Note that the metaphors are intended, *not* in any way as interface metaphors, but to support the evaluator in a focused study of how well certain important aspects of human thinking are taken into account in the user interface under inspection.

Metaphor of Habit Formation. Habits are shaping most of our thought activity and behaviour—e.g. as physical habits, automaticity, all linguistic activity, and habits of reasoning. *The metaphor is*: Habit formation is like a landscape eroded by water. This metaphor should indicate how a person's formation of habits leads to more efficient actions and less conscious effort, like a landscape through erosion adapts for a more efficient and smooth flow of water. Creeks and rivers will, depending on changes in water flow, find new ways or become arid and sand up, in the same way as a person's habits will adjust to new circumstances and, if unpracticed, vanish.

In design, there is an abundance of examples of user interfaces that violate human habits. One example is adaptive menus, used for example in Microsoft Office 2000. Adaptive menus change the layout of the menu according to how often menu items are used, for example by removing or changing the position of items seldom used. However, adaptive menus make it impossible to form habits in the selection of menu items [13], since their position may be different from when they were previously selected. A study by Somberg [15] showed the efficiency of constant position placement of menu items compared to menus that change based on use frequency. Somberg, however, did not explicitly link habit formation to the usefulness of constant placement of menu items. Note that the common practice of adding a fixed number of, say, recently used files or fonts to the bottom or top of a menu does not interfere with habit formation and may decrease time taken to select a menu item [14].

In usability inspection this metaphor calls for considering: Are existing habits supported? Can effective new habits, when necessary or appropriate, be developed? Can the user use common key combinations? Is it possible for the user to predict, a requisite for forming habits, the layout and functioning of the interface?

Metaphor of the Stream of Thought. Human thinking is experienced as a stream of thought-in the continuity of our thinking, the richness and wholeness of a person's mental objects, of consciousness, and subjective life. *The metaphor is*: Thinking as a stream of thought. This metaphor was proposed by William James [7] (vol. I, p. 239) to emphasize how consciousness does not appear to itself chopped up in bits: 'Such words as "chain" or "train" do not describe it fitly. It is nothing jointed; it flows'. Particular issues can be distinguished and retained in a person's stream of thought with a sense of sameness, as anchor points, which function as 'the keel and backbone of human thinking' [7] (vol. I, p. 459).

In design, a simple, yet effective, attempt to recreate part of the richness of the stream of thought when users return to resume interrupted work, is Raskin's design of the Canon Cat [13]. When the Canon Cat is started, the display immediately shows up as it was before work was suspended. Not only does this allow the user to start thinking about the task at hand while the system is booting. It also provides help in remembering and recreating the stream of thought as it was when work was interrupted.

In usability inspection this metaphor calls for considering: Is the flow in users' thought supported in the interface by recognizability, stability and continuity? Does the application make visible and easy accessible such interface elements that relate to the anchor points of users' thinking about their tasks? Does the application help users to resume interrupted tasks?

Metaphor of the Dynamics of Thinking. Here is considered the dynamics of human thinking, the awareness shaped through a focus of attention, the fringes of mental objects, association, and reasoning. *The metaphor is*: Awareness as a jumping octopus in a pile of rags. This metaphor was proposed by Peter Naur [9] (pp. 214-215) to indicate how the state of thought at any moment has a field of central awareness, that part of the rag pile in which the body of the octopus is located; but at the same time has a fringe of vague and shifting connections and feelings, illustrated by the arms of the octopus stretching out into other parts of the rag pile. The jumping about of the octopus indicates how the state of human thinking changes from one moment to the next.

In design, modal dialog boxes prevent the user from switching to potentially relevant information—in Microsoft Word, for example, it is not possible to switch back to the document to look for a good file name once the 'save as ...' dialog has began.

In usability inspection this metaphor calls for considering: Are users' associations supported through flexible means of focusing within a stable context? Do users associate interface elements with the actions and objects they represent? Can words in the interface be expected to create useful associations for the user? Can the user switch flexibly between different parts of the interface?

Metaphor of the Incompleteness of Utterances. Here is focused on the incompleteness of utterances in relation to the thinking underlying them and the ephemeral character of those utterances. *The metaphor is*: Utterances as splashes over water. This metaphor was proposed by Naur [9] (pp. 214-215) to emphasize how utterances are incomplete expressions of the complexity of a person's current mental object, in the same way as the splashes tell little about the sea below.

For design, one implication of the metaphor of utterances as splashes over the water is that we must expect users to describe the same objects and functions incompletely and in a variety of ways. Furnas et al. [4] investigated the diversity in words used for describing commands and everyday objects. On the average, two participants described the same command or object by the same term with less than 20% probability. The most popular name was chosen only in 15-35% of the cases. Furnas et al.'s suggestion for relieving this problem is called the unlimited alias approach. Instead of using a fixed set of words for commands and functions, the unlimited alias approach lets users enter any term they want. If the term is not in the range of terms initially suggested by the designer of the system—which the data of Furnas et al. and the metaphor suggest it often will not be—the system may interactively suggest appropriate commands or object names. This approach is coherent with the metaphor and uses interactivity to clarify the intentions of the user. On the other hand, the approach partly goes against the metaphor of habit formation.

In usability inspection this metaphor calls for considering: Are changing and incomplete utterances supported by the interface? Are alternative ways of expressing the same information available? Are the interpretations of users' input in the application made clear? Does the application make a wider interpretation of users' input than users intend or are aware of?

Metaphor of Knowing. Human knowing is always under construction and incomplete. *The metaphor is*: Knowing as a site of building in progress. Also this metaphor was proposed by Naur [9] (p. 214-215) and meant to indicate the mixture of order and inconsistency characterizing any person's insight. These insights group themselves in many ways, the groups being mutually dependent by many degrees, some closely, some slightly. As an incomplete building may be employed as shelter, so the insights had by a person in any particular field may be useful even if restricted in scope.

In design, mental models have been extensively discussed. Consider as an example Norman's [12] description of the use of calculators. He argues that the use of calculators is characterized by users' incomplete understanding of the calculators, by the instability of the understanding, by superstitions about how calculators work, and by the lack of boundaries in the users' understanding of one calculator and another. These empirical observations by Norman are coherent with the ideas expressed by the metaphor of knowing.

In usability inspection this metaphor calls for considering: Are users forced by the application to depend on complete or accurate knowledge? Is it required that users pay special attention to technical or configuration details before beginning to work? Do more complex tasks build on the knowledge users have acquired from simpler tasks? Are users supported in remembering and understanding information in the application?

Further examples. In [3], each of the metaphors and their implications for user interfaces are described in more detail. In [5], we propose the metaphor-based usability inspection technique and discuss how to conduct such a usability inspection. In [6], we present initial empirical results on the effectiveness of the inspection technique.

3. CONCLUSION

General properties of thinking activity known to all of us by introspection were emphasized through five metaphors, which build upon the work of William James and of Peter Naur. The metaphors catch psychological aspects of habit formation, stream of thought, awareness, utterances, and knowing. With the possible exception of awareness, these aspects of human thinking are rare in recent years of HCI literature (cf. [3]). From commonly available user interfaces and from a selection of empirical studies, the utility of the metaphors was illustrated by their ability to clarify designs and notions in HCI. We suggest that the metaphors, by virtue of their psychological recognizability and focus on basic aspects of thinking, can help designers consider important human traits.

When using MOT as a usability inspection technique, inspection is focused on the users' mental activity through the five metaphors of essential aspects of human thinking.

In an experiment [6], MOT compared to Heuristic Evaluation (HE) uncovers more of the usability problems that were assessed severe on users and complex to repair. In addition, the evaluators using MOT show a stronger agreement by finding the same problems more often; and evaluators use less time to perform their evaluation.

It is remarkable how MOT in its first experimental study has given good results compared to HE, the usability inspection technique most widely used in industry. HE usually performs very well in comparison with other inspection techniques, e.g. cognitive walkthrough and GOMS-based techniques.

It must be emphasized that these results are preliminary and have to be challenged by further studies. What happens when MOT is used for evaluating interfaces in non-traditional use contexts, when the evaluators are more proficient, or when MOT is used in design work? In the experiment, however, usability inspection by metaphors of human thinking showed to be viable.

4. REFERENCES

[1] Baars, B. J. *A Cognitive Theory of Consciousness*, Cambridge University Press, Cambridge, 1988.

[2] Baars, B. J. *In the Theater of Consciousness*, Oxford University Press, New York, 1997.

[3] Frøkjær, E. & Hornbæk, K. Metaphors of human thinking in HCI: habit, stream of thought, awareness, utterance, and knowing, in *Proceedings of HF/OzCHI 2002* (Melbourne, Nov 2002).

[4] Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. The vocabulary problem in human-system communication, *Communications of the ACM*, *30*, 11 (1987), 964-971.

[5] Hornbæk, K. & Frøkjær, E. Evaluating user interfaces with metaphors of human thinking, in *Proceedings of ERCIM Workshop on User Interfaces for All* (Chantilly, France, Oct 2002).

[6] Hornbæk, K. & Frøkjær, E. Usability inspection by metaphors of human thinking compared to heuristic evaluation, 2002, under review.

[7] James, W. *Principles of Psychology*, Henry Holt & Co., 1890.

[8] Naur P., Human knowing, language, and discrete structures, in *Computing: a Human Activity,* ACM Press/Addison Wesley, 1988, 518-535.

[9] Naur, P. *Knowing and the Mystique of Logic and Rules*, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1995.

[10] Naur, P. CHI and human thinking, in *Proceedings of NordiCHI 2000* (Stockholm, Oct 2000),

[11] Naur, P. *Anti-Philosophical Dictionary*, naur.com Publishing, 2001.

[12] Norman D., Some observations on mental models, in Gentner, D. and Stevens, A. L. *Mental Models,* Erlbaum, 1983, 7-14.

[13] Raskin, J. *The Humane Interface: New Directions for Designing Interactive Systems*, Addison-Wesley, Reading MA, 2000.

[14] Sears, A. & Shneiderman, B. Split menus: effectively using selection frequency to organize menus, *ACM Transactions on Computer-Human Interaction*, *1*, 1 (1994), 27-51.

[15] Somberg, B. L. A comparison of rule-based and positionally constant arrangements of computer menu items, in *Proceedings of CHI+GI'87* (Toronto, Canada, Apr 1987), ACM Press, 255-260.

Combining Co-Located and Distributed Collaboration Tools

Klaus Marius Hansen and Christian Heide Damm
Centre for Pervasive Computing
Department of Computer Science, University of Aarhus
Aabogade 34, 8200 Aarhus N, Denmark
{marius, damm}@daimi.au.dk

ABSTRACT

This extended abstract reports on our ongoing efforts in combining co-located and distributed collaboration tools. Co-located collaboration tools are well-researched and can support effective collaboration between collaborators in the same location. Distributed collaboration are also well-researched and support physically separated collaborators. Their combination, however, is not well-understood and issues include physical versus virtual group interaction and coordination and the need for the integration of different interface paradigms.

1. INTRODUCTION

Organizations are becoming increasingly distributed, and at the same time work is becoming more complicated [2] yielding an increased need for both computer support of distributed work processes and of co-located, collaborative work processes. Consider the following scenarios, in which a combination of co-located and distributed collaboration is needed.

> Chris and Bob are building an add-on to an existing Enterprise Resource Planning (ERP) system. They are now modelling how to integrate their add-on into the ERP system, but the company expert on the ERP system is currently away on a business trip. Thus, they call the expert and try to guide him through their problems while they are adapting his answers to their current situation.

Here, a group of novices collaborate with a remote expert that cannot be continuously available on-site.

> Peter and Keith are working on a customer-service system for a globally distributed shipping company. The problem domain is extremely complex, and the knowledge of it is distributed among a number of domain experts spread around the globe. Peter and Keith are modelling the part of the system pertaining to the booking process, and in order to get input on how booking is done in the East, they call and write e-mails to a designated customer service agent in Japan.

In this scenario, a remote domain expert helped a group of developers doing problem domain modelling. Other types of situations in which these types of scenarios occur include teaching, mentoring, and general peer interaction if a peer in a group needs to work remotely.

We are building co-located and distributed collaboration tools for object-oriented modelling in the Unified Modeling Language (UML) [5] in order to support scenarios as these:

1.1 The Knight Tool

The *Knight tool* [4] supports synchronous, co-located object-oriented modelling through gesture-based interaction suited for, e.g., interactive electronic whiteboards. Figure 1 shows the typical setup to the left and an example of using gestures to the right.

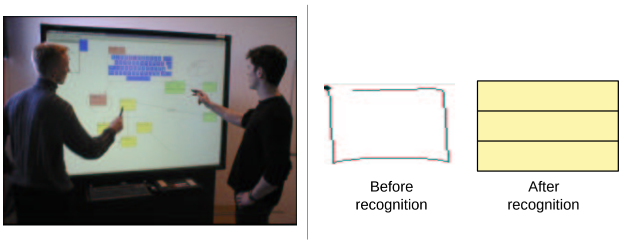


Figure 1: Left: Use of Knight on an Electronic Whiteboard. Right: Recognition of Pen Strokes in Knight

The Knight Tool has been commercialised as *Ideogramic UML* by Ideogramic ApS (http://www.ideogramic.com/).

The *Distributed Knight tool* is an extension for the Knight tool created in order to support synchronous, distributed collaboration. The research goals in creating this are two-fold: first, to experiment with a novel peer-to-peer architecture for distributed collaboration tools based on the publish/subscribe paradigm for distributed communication [3], and, secondly, to experiment with interaction issues in combining co-located and distributed collaboration.

2. DISCUSSION

[7] introduced the idea that tools for co-located collaboration, or *Single-Display Groupware*, may be a way to engage multiple, co-present users by, e.g., *enabling types of interaction that require multiple users*, *enriching existing collaboration at a computer*, or *encouraging peer-learning and peer-teaching* (Figure 2). Equipping a traditional desktop computer with two mice may enable this type of collaboration. Electronic whiteboards is another technology that inherently supports single-display collaboration.

These kinds of applications take advantage of the efficiency of co-located, physical collaboration that is much more effective than dis-

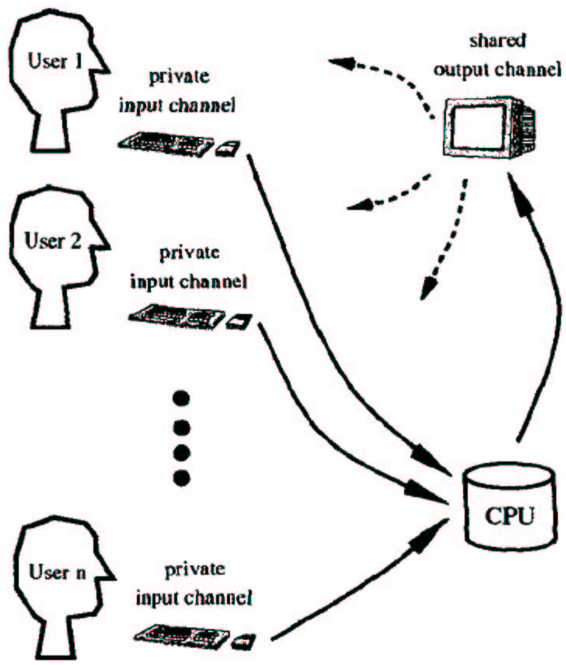


Figure 2: Single-Display Groupware Architecture

tributed collaboration due to more effective verbal and nonverbal communication [6]. On the other hand, distributed collaboration is sometimes necessary, a.o. for the reasons detailed above. One goal of distributed collaboration applications is to make coordination and communication as "natural" as possible through the use of, e.g., real-time updates, full duplex audio, and video [1].

The combination is the cause for potential conflicts between the natural support for collaboration in the co-located situation and the virtual collaboration support attempts in the distributed situation. In particular, issues include:

- *Differing interface paradigms.* In our case, one end of the distributed session uses a standard desktop computer and the other uses an electronic whiteboard. Different kind of interaction paradigms are suited for each kind of device. How do you combine these and provide suitable interaction mechanisms on each side?

- *Rich versus poor communication channels.* Co-presence provides a rich set of communication possibilities: sounds, gazes, gestures, nods, ... In distributed collaboration, this is often simulated using audio and video connections. How will the absence of rich communication possibilities between both sides mix with the rich communication possibilities of the one side?

- *Physical versus virtual group interaction*. Co-located persons use physical artefacts and affordances when collaborating: pens on a whiteboard are, e.g., used to signal focus of interest by pointing or for coordinating turn-taking. How does these kinds of physical group interaction relate to the virtual group interaction of distributed collaboration?

We plan to address and examine these issues through a series of experiments:

3. EXPERIMENTS

Our initial investigation of the combination of co-located and distributed collaboration tools will be focussed on taking a functioning tool of each kind and then combining them. To do this, we are currently setting up a number of user studies. We plan on doing four studies of collaborating groups, with three persons in each group. In each study, two persons will be working with Knight on an electronic whiteboard, and one person will be working with Knight on a desktop computer. These nodes will the run Distributed Knight and connect to collaborate on a problem. The groups will collaborate on a problem of their own design, and we imagine that two groups will be of the novices-expert type and that two groups will be of the developers-domain expert type.

Each group will be observed and possibly videotaped by two persons that will focus on the communication and coordination that takes place when using Distributed Knight. In particular, we will focus on the interplay between co-located and distributed collaboration.

4. SUMMARY

We have reported from our initial thoughts about and our experiments with combining co-located and distributed collaboration tools. Currently, we have two technologies, a co-located and a distributed Knight, which work effectively in isolation. Our current interest is then to combine these, conduct use studies, and investigate a range of issues pertaining to this combination.

5. REFERENCES

[1] M. Beaudouin-Lafon, editor. *Computer Supported Cooperative Work*. Wiley, 1999.

[2] CNN. Telework trend rooted in convenience. http://www.cnn.com/2001/TECH/ptech/11/12/telework-.comdex.idg/index.html, 2001.

[3] C.H. Damm and K.M. Hansen. Distributing Knight – Using type-based publish/subscribe for building distributed collaboration tools. In Proceedings of NWPER 2002, 2002.

[4] C.H. Damm, K.M. Hansen, and M. Thomsen. Tool support for object-oriented cooperative design: Gesture-based modeling on an electronic whiteboard. In *Proceedings of CHI 2000, ACM Conference on Human Factors in Computing Systems*, pages 518–525, 2000.

[5] OMG. Unified Modeling Language specification 1.4. Technical Report formal/01-09-67, Object Management Group, 2001.

[6] R.B. Smith, T. O'Shea, E. O'Malley, E. Scanlon, and J. Taylor. Preliminary experiments with a distributed, multimedia problem solving environment. In *Proceedings of EC-CSCW'89*, pages 19–34, 1989.

[7] Jason Stewart, Benjamin B. Bederson, and Allison Druin. Single display groupware: a model for co-present collaboration. In *Proceeding of the CHI 99 conference on Human factors in computing systems : the CHI is the limit*, pages 286–293. ACM Press, 1999.

Structuring the User Interface Design Process - First Steps Towards a Frame-of-Reference for User Interface Design Issues

Morten Borup Harning
Dialogical ApS
Inavej 30
DK-3500 Værløse
+45 4447 4427
harning@dialogical.dk

ABSTRACT

This paper introduces the frame of reference used in the design method called Software User Interface Engineering (SUIE) [2]. The frame of reference enables the designer to structure the user interface design work.

Keywords

User interface design, design method, conceptual design, direct manipulation

1. INTRODUCTION

Most research in user interface *design* within the field of human-computer interaction tends to focus on task analysis (e.g. [3]) or techniques for acquiring different kinds of contextual information [1] or on technical innovations e.g. within fields like tangible user interfaces. Turning all of the acquired information into a concrete user interface, including selecting what information to be shown, and how to layout and present information and related interaction techniques, are however seldomly described - leaving some of the most important parts of the user interface design to "magic" or intuition. Often the only guidance provided is descriptions of relevant knowledge within a broad range of related fields, such as psychology of perception, graphical design, cognitive science etc. (e.g. [4])

Software User Interface Engineering [2] was proposed to bridge this gap between knowledge of human factors and the actual user interface design, that needs to be part of the software development process, but nevertheless receives little attention in the software engineering literature.

Software User Interface Engineering (SUIE) was developed at Copenhagen Business School in the period from 1991-1996. The main focus at that time was to come up with a design method, that could help designers design better user interfaces.

The method built heavily on the ideas of direct manipulation ([6], [7], [10] and [11]) showing how these principles could be applied - even on platforms known not to support direct manipulation, such as interactive web-systems. The second strong influence was the principles of conceptual design (e.g. [9]) but also [8].

Having completed the design method it became clear, that work behind the method fell into several groups:

1. Notations used to describe design
2. Process – the recommended order of design activities
3. Frame of reference structuring the design issues

All closely related but nevertheless separate groups.

Traditionally design methods focus on (1) and (2) the notations used and the recommended sequence of the design activities. Some design methods even claim to be independent of the frame of reference implied by the process and notations.

This paper first briefly describes the design process proposed by SUIE and then presents the frame of reference implied.

2. PROCESS

SUIE divides the design process into eight closely related design step. Figure 1 illustrates the relationships between the steps as well as the suggested order of the steps.

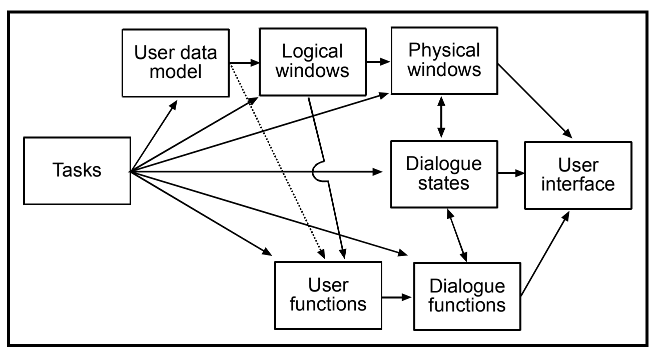


Figure 1. The design steps found in SUIE and the relationships between each of the steps.

All design activities are focused around "tasks", which refers to all available contextual information e.g. as described in [1]. The central part of the design method is the design of a so-called user data model, which captures the design of the concepts to be used. Following the design of the user data model is the designs of logical windows focusing on how information should be presented in order to best support the users tasks. The user functions covers all semantic functionality, including the functionality necessary to manipulate the information presented in the logical windows. Finally the design of dialogue states and dialogue functions describes how the detailed dialogue should be implemented. Steps similar to the user data model, the dialogue states and dialogue are found in a number design methods, while the combination is unique to SUIE.

3. FRAMEWORK

As mentioned in the introduction SUIE can be seen as a frame of reference, that can be used to structure the design issues needing to be addressed in a given user interface design process. The SUIE framework consists of five main groups of information: Contextual information, Conceptual design, Logical design, Dialogue design and Physical design. The steps are closely related to the design activities outline in section 2, but makes sense out of that context as well. The five groups are:

1. Contextual information
2. Conceptual design
3. Logical design
4. Dialogue design
5. Physical design

Contextual information includes scenarios, task descriptions and any other contextual information, that might influence the design.

Conceptual design includes information concerning the information associated with a given concept, about how the concept will appear in the interface, naming conventions etc. The conceptual design is closely related to the data or object model found in the technical design, but does not include descriptions of functionality (often included in object models).

The *logical design* includes descriptions of the information artefacts (logical windows) and semantic functions (user functions) required to satisfy the information need of the user.

Dialogue design includes the descriptions of all possible dialogue states and dialogue functions, including how each of the semantic function is implemented and how they affect the current dialogue state.

Physical design refers to the detailed design of the user interface, either as a prototype including detailed layout, graphical design or as specifications sufficiently detailed to form the basis for the implementation.

SUIE uses a number of proprietary notations to capture the design decisions related to each of the five groups (user data models, logical windows, user function diagrams and dialogue state diagrams). The notations have been designed to facilitate a user oriented design process – especially in the early faces of the design. They were designed to capture most of the central design decisions but will never be able to capture all of the actual design decisions made. A number of different versions of the notations have been used throughout the evolution of the method. They capture different design issues, often more or less overlapping. That is one of the reasons why a good understanding of the underlying frame of reference is important understand the strength and weaknesses of the design method.

4. CONCLUSION

Based on the design method called Software User Interface Engineering it seems possible to infer the underlying frame of reference. The current version of the frame of reference has shown to capable of explaining a number of important design decisions, e.g. made it possible to decompose existing designs. A possible next step will be to investigate how well the frame of reference can explain design problems, e.g. found in usability tests.

5. REFERENCES

[1] Beyer, H. and Holtzbaltt, K. (1998) *Contextual Design – Defining Customer-Centred Systems*. San Francisco, California, Morgan Kaufmann Publishers.

[2] Harning, M.B. *Software User Interface Engineering (SuiE) - a structured method for designing user interfaces* (in Danish), Ph.D. dissertation, Faculty of Economics, Copenhagen Business School, Ph.D.-series 5.97, 1997

[3] Lim, K.Y. and Long, J. (1992) *The MUSE Method for Usability Engineering*. Cambridge Series on Human-Computer Interaction, Cambridge University Press.

[4] Preece, J. et.al. (1994) *Human-Computer Interaction*. Wokingham, Addison-Wesley.

[5] Harning, M.B., *An Approach to Structured Display Design - Coping with Conceptual Complexity*, Proceedings of CADUI'96

[6] Shneiderman, B., *Direct Manipulation: A Step Beyond Programming Languages*, Computer, August, 1983

[7] Hutchins, E., Hollan, J., Norman, D., *Direct Manipulation Interfaces*, Published in Norman & Draper (eds), "User Centred System Design: New Perspectives on Human-Computer Interaction", Hillsdale, 1986.

[8] Norman, D., *The Psychology of Everyday Things*, Basic Books, 1988.

[9] Foley, J., van Dam, A., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1982.

[10] Ahlberg, C., Williamson, C. and Shneiderman, B. Dynamic Queries for Information Exploration: An Implementation and Evaluation, Proceedings of CHI'92.

[11] Ahlberg, C., and Truvé, S., *Tight Coupling: Guiding User Actions in a Direct Manipulation Retrieval System*, Proceeding of HCI'95.

Requirements Specifications and Scenarios: Two Design Artefacts in Software Engineering

Morten Hertzum
Centre for Human-Machine Interaction
Risø National Laboratory
DK-4000 Roskilde, Denmark
+45 4677 5145
morten.hertzum@risoe.dk

1. INTRODUCTION

Requirements specifications are an established element of software-engineering projects, and scenarios have gained acceptance in both research and practice as a way of grounding projects in the users' work. However, the research on requirements specifications and scenario-based design includes very few studies of how such design artefacts are actually used by practising software engineers in real-world projects. This study [3, 4] investigates how a requirements specification and a set of scenarios entered into defining how software engineers and users envisioned the future interaction between tasks, users, and the system under development.

The company where the study took place is a large software house, which has developed and marketed a range of systems for use in municipal institutions. The studied project concerns a system to support municipal authorities in the handling of cases concerning child support and alimony (CSA). The CSA project is to completely redevelop the company's existing CSA system, which has been in operation for almost two decades. The CSA project is staffed with 17 people with an average of more than ten years of professional experience, and the project will, according to the project plan, last three years.

The data collected for this study cover the first year of the project and comprise attendance at the two-day start-up seminar, observation of the fortnightly project status meetings, interviews with core project participants, and inspection of project documentation. The meetings and interviews were recorded on tape and transcribed. The documents, which provide evidence of the evolution and intermediate outcomes of the project, include among other things the final as well as several preliminary versions of the requirements specification and the scenarios.

2. REQUIREMENTS SPECIFICATION

The requirements specification consisted of 221 requirements, which were maintained as individual entries and organised by means of a classification scheme. The initial purpose of the requirements specification was to facilitate communication with the user representatives during the requirements-engineering process. After its completion the requirements specification assumed a double role of, on the one hand, contract between users and development organisation and, on the other hand, checklist for the CSA engineers during the development and evaluation of subsequent design artefacts. In these roles, the requirements specification and its classification scheme had a primarily indirect effect on the design process. For example, the scenarios were not generated from the requirements specification. Rather, they were developed on the basis of the CSA engineers' knowledge of the domain and the users' tasks, supplemented by discussions with the user representatives and some reading of CSA legislation. The requirements specification was used most visibly when it was brought in at selected points in the process, for example to validate that design artefacts such as the scenarios met the full range of requirements. However, the requirements specification also affected the design process in another, more fundamental way as a constituent part of the assumptions about the scope of the project.

The requirements specification for the new CSA system inherited a lot of its structure from the existing CSA system. This introduced a potentially undue bias toward preserving existing system facilities and ways of working. The CSA engineers were aware of this risk but explicitly argued that it was more important that the requirements classification depicted the world in a way recognisable to the user representatives. While this is convenient, it also illustrates how the requirements classification indirectly constrained the requirements-engineering process to requirements that could be conceived of within the framework of the existing system [see also 1]. This is apparently at odds with the activities undertaken to facilitate the user representatives in an open-ended search for the optimal balance between tradition and transcendence (e.g., a vision workshop conducted as part of one of the meetings). The CSA engineers were, however, faced with two contradictory concerns. On the one hand, they needed to conduct the requirements-engineering process in a way that honoured expectations of adequate user involvement. On the other hand, they needed to maintain some level of control over the direction, scope, and outcome of the requirements-engineering process, which concluded in a specification of what the customers had requested and the developers agreed to deliver – a contract. The requirements classification played a discrete but important role in the CSA engineers' handling of these two concerns in that it enabled the CSA engineers to act in accord with expectations of adequate user involvement while at the same time constraining the process. On several occasions, the CSA engineers explicitly asked the user representatives for new ideas and visions regarding the system but, at the same time, the meetings with the user representatives evolved around a walkthrough of the classified requirements, one category at a time. Under these circumstances, the user representatives had few ideas for new facilities that would enhance the system.

The tension between open-ended user involvement and the contractual aspect of requirements specification was rooted deeply in the CSA engineers' perception of their work, and they considered disregard of this tension tantamount to being unprofessional. This was, for example, a problem in their relations with a usability specialist who considered it her role to systematically "adopt the users' perspective". To the CSA engineers handling these conflicting interests was normal, natural practice [2] to the extent that they probably remained largely unaware of how effective the requirements classification was as a means of controlling the scope of their project.

3. SCENARIOS

The scenarios were schematised descriptions of the courses of activities that constitute CSA work. The grounding in the flow of CSA work means that the scenarios are rich in the information needed in the day-to-day management of CSA cases, such as how activities are sequenced, what triggers them, and when they trigger other activities. This means that the scenarios make the users' work recognisable to the CSA engineers as a complex but organised human activity.

Each scenario consists of a chronological progression of activities. Typically, CSA work progresses continually for only brief intervals of time; then further progress must, for example, await that the person entitled to receive CSA supplies additional information. Consequently, most of the steps in the scenarios are triggered by events. These events define the information that must be provided before further progress can be made or they lead to the execution of subtasks that are only relevant when certain conditions occur. Consequently, the scenarios preserve the real-world ordering of the activities involved in performing a task and also delineate the events or circumstances that affect whether and when various activities are performed. The CSA engineers perceived the scenarios as quite coherent descriptions of CSA tasks and considered this a valuable and distinguishing feature of the scenarios.

When the scenarios were discontinued to free key CSA engineers for other project activities several of the CSA engineers were concerned that the discontinuation of the scenarios would deprive them of valuable information about the various aspects of CSA work. This concern was partly an appreciation of the scenarios and partly instigated by the common impression among the CSA engineers that the other design artefacts did not provide them with an equally good tool for understanding CSA work. What the CSA engineers lost with the scenarios was a design artefact that aimed at describing the users' work as tasks consisting of a structured sequence of interrelated activities. Contrary to the scenarios, the requirements specification can best be characterised as an extensive list enumerating large amounts of separate details. The requirements specification provides no information about how the 221 requirements impact on each other. It is, for example, left entirely to the reader of the requirements specification to determine whether it contains conflicting requirements.

The scenarios were developed as a tool for the stakeholders internal to the CSA project. The descriptive nature of the scenarios made them accessible to all CSA engineers and meant that the scenarios were not biased toward, or owned by, a subgroup of CSA engineers responsible for a specific part of the project. Further, all CSA engineers considered it natural to relate their work to the users' tasks, which were the common referent of the scenarios. This can be illustrated by some of the uses to which the scenarios were put. The scenarios generated a number of the events and elementary processes, which made up the business model, and they were a defining input in the development of the dialogue flow of the user-interface prototype. In addition, the CSA engineers preferred the scenarios as their base representation in a joined effort to establish the status of their project after six months had elapsed.

Johnson-Laird and Wason [5] have vividly illustrated people's superiority in dealing with concrete descriptions of real-world affairs, as opposed to abstract descriptions. Whereas abstract descriptions tend to be experienced as logical puzzles, concrete descriptions of real-world affairs seem to tie in with people's general abilities to deal with their world and to be experienced as much more straightforward. Thus, the coherence and concrete, real-world feel of the scenarios may be distinct advantages, which made the CSA engineers better able to grasp CSA work and reason about the suitability of different design ideas.

4. CONCLUSION

It is inherently difficult for people to transcend their current way of perceiving things and envision how tasks, users, and technology should interact in constituting the future use situation. Design artefacts, such as requirements specifications and scenarios, may affect this complex process in very different ways and, thus, play different roles in software-engineering projects.

5. ACKNOWLEDGEMENTS

This work has been supported by the Danish National Research Foundation through its funding of the Centre for Human-Machine Interaction. Special thanks are due to the members of the CSA project group who have put up with my presence in spite of their busy schedule.

6. REFERENCES

[1] Bowker, G.C. & Star, S.L. (1999). *Sorting Things Out: Classification and Its Consequences*. Cambridge, MA: MIT Press.

[2] Garfinkel, H. (1967). "Good" organizational reasons for "bad" clinic records. In H. Garfinkel, *Studies in Ethnomethodology*, pp. 186-207. Englewood Cliffs, NJ: Prentice Hall.

[3] Hertzum, M. (2002). Making use of scenarios: A field study of conceptual design. Submitted for publication.

[4] Hertzum, M. (2002). Small-scale classification schemes: A field study of requirements engineering. Submitted for publication.

[5] Johnson-Laird, P.N. & Wason, P.C. (1977). A theoretical analysis of insight into a reasoning task. In P.N. Johnson-Laird & P.C. Wason (eds.), *Thinking: Readings in Cognitive Science*, pp. 143-157. Cambridge, UK: Cambridge University Press.

Context based information behaviour and social interaction

Jette Hyldegård
Associate professor, ph.d.student
Royal School of Library and Information Science
Birketinget 6, 2300 Kbh. S.
+45 32 58 60 66
jh@db.dk

General Terms
Human Factors.

Keywords
Information behaviour, collaborative information seeking, social interaction, qualitative study, diaries.

1. INTRODUCTION
Information seeking and information systems is an important and integrated part of work domains and work practices which directly or indirectly determines and presumes the quality of work and the cooperative activities among workers. While many different models of information seeking have been proposed, implicit in most of them is the assumption that the information seeker is an individual. Recently, researchers have begun to challenge this assumption by exploring the social, contextual and collaborative dimensions of information behaviour and information seeking. This paper presents a qualitative study of the information behaviour of eight library- and information science students during the writing of an examination paper. The aim is to explore the social aspect of information behaviour and how behaviour might differs between students working alone and in groups and even between group members. The study was conducted in spring 2002 and is part of a ph.d.-project on context based information behaviour and social interaction.[1] Based on the results from this study, a larger study in a real work setting will be carried out in future. The aim of the ph.d.-project is to contribute to the theory of information behaviour in context that may qualify the design of information systems and human-computer-interaction within the concept of user modelling, information filtering and social navigation ((to fit the problems and tasks of work).

[1] The ph.d. project is carried out at The Royal School of Library and Information Science and runs from August 2001 to July 2005.

2. INFORMATION BEHAVIOUR IN CONTEXT
Though providing a user centered approach to studying and understanding information behavior, the cognitive viewpoint in information science has fundamentally focused on attributes of the individual, i.e. to understand the cognitive and emotional motivations for information behavior that carry across context or are independent of context. This is in contrast to the social cognitive viewpoint where context (particularly attributes of the social and organizational context) becomes the focus for understanding information behavior. Context are frameworks of meaning and describes the collection of events, histories, culture, knowledge and understanding, which exists together at a point in time [4]. Different levels of context may be described, e.g. a collaborative information seeking context and a organisational context [2]. Context may again consists of a variety of situations that denotes the dynamic environments within which interpretive processes unfold, become ratified, change and solidify [4]. Situations are where information behavior takes place; the dynamic aspect of context. While different definitions of information behavior is used among researchers, in this study it refers to the study of how people need, seek, give and use information in different contexts. The context of this study is the writing of an examination paper, due to a course on document- and knowledge management systems within the masters program of library and information science (LIS). Following the work of Kuhlthau [1] the study was based on six different stages of the search process during a two months period of time each showing different attributes of the individual depending on where in the search process he or she was. For example Kuhlthau identified feelings such as uncertaincy, confusion, optimism, frustration, relief and satisfaction over time, attributes which also have been validated by Vakkari [3]. To elaborate on Kulthaus model which was context independent and focusing on the individual information seeker this study would explore if or how social and collaborative aspects might affect the information behavior and related attributes of the group of students.

3. QUALITATIVE STUDY OF INFORMATION BEAVIOUR AND SOCIAL INTERACTION AMONG LIS-STUDENTS

3.1 Participants

Eight master students in library and information science participated, three individuals (working alone on the paper) and two groups. They all had a bachelor in library and information science and used information systems on a frequently basis. They had different experiences with group working based on earlier writing projects.

3.2 Data collection and analysis

The study is based primarily on qualitative data. After an introduction to the study each participant was provided with a two-page diary form to be filled out electronically on a daily basis and send to the researcher on a weekly basis. They should note all the tasks and related activities carried out during a day due to the examination paper, the amount of time spent on each activity, which information or communication sources they might have been using (e.g. persons, books, articles, databases, telephone, e-mail) and finally indicate the perceived degree of feelings such as uncertaincy, frustration, satisfaction, if recognized with a number from 1-5. To judge from the students comments on the use of diaries it seemed that this data collection method also served as a tool for reflection on information seeking and work practice. The diaries formed also the basis for three semi-structured interviews which were conducted with each participant in the beginning of , during and after the writing of the paper. For both groups of participants (individual and group members) the interviews focused on their understanding and perception of the goal or problem at hand (the writing of the paper), the related information behavior activities, for example the importance and use of information resources and persons, such as group members or the supervisor during different stages of the project period. But in the interviews with the group members the focus was also on identifying group related attributes, such as collaboration and roles among participants that might limit or facilitate information behavior and collaborative activities in certain ways and over time. Furthermore the interviews tried to explore if or how perceptions of situations and context were shared among group members that might again influence the information behavior activities.

Each interview lasted for about 45 minutes and were recorded on tape. Since the beginning of the study was a bit delayed in proportion to the starting point of the project the students were asked to fill out a questionnaire in the beginning of the study which tried to catch up un their information behavior in the time between, besides getting some demographic information. The diaries have been printed out and the interviews have been transcribed. Right now the data is being analyzed, so no conclusions can yet be made. Besides the aim of the study mentioned above, the hope is to explore how dimensions of social interaction affects information behavior to further strengthen and define the research problem of the ph.d.-project.

4. RESEARCH QUESTIONS - DISCUSSION

Though no findings can be presented right now some vague pattern seems to emerge. It seems that the motivation for social interaction is determined by the task, the stage in the process and psychological factors. The information behavior among group members seem to have both the intention of searching, exchanging and distributing information – not only to share the information, but also to communicate to the other group members the interests or preferences of the individual. The feelings identified by Kuhlthau in relation to different stages over time also seem to have been identified in this study, but still we need an answer to the question if or how the affective dimension of information behavior is related to social factors and then might differ between individual and group member participants. But instead of focusing on either individuals or groups (socially bound members) we may focus on the dynamics of social interaction within information behavior and collaborative information seeking based on the context and situation given.

- How is information behavior and social interaction related
- What is the dynamics of collaborative information seeking
- Which methods will support the exploration of information behavior and social interaction; how will diaries support the collection of relevant data

5. REFERENCES

[1] Kuhlthau, C.C. Inside the Search Process: Seeking Meaning from the Users Perspective. Journal of the American Society for Information Science, vol. 42, No. 5, (1991), 361-371.

[2] Prekop, P. A Qualitative Study of Collaborative Informaiton Seeking. Journal of Documentation, vol. 58, No. 5, (2002), 533-547.

[3] Vakkari, P. Cognition, Sources and Contributory Information of Documents in Writing Research Proposals: A Longitudinal Case Study. Kraft, D. (Ed). ASIS 2000: Proceedings of the American Society for Information Science (ASIS) 63rd Annual Meeting: vol. 37, 2000, November 16-20, Chicago, IL. Medford, Nj: Information Today, Inc. for ASIS, 2000, 352-362.

[4] Sonnenwald, D. Evolving Perspectives of Human Behaviour: Contexts, Situations, Social Networks and Information Horizons. In: Wilson, T.D.; Allen, D.K. (eds). Exploring the contexts of Information Behaviour: proceedings of the 2nd. International Conference on Research in Information Needs, Seeking and Use in Different Contexts; 1998, August 13-15; Sheffield, Uk. London, UK: Taylor Graham, 1999. 176-190.

Proposal for Verbal Attributes of Musical Sounds

Kristoffer Jensen
Music Informatics Laboratory
Department of Datalogy, Universitetsparken 1
2100, Copenhagen
+45-35321400, krist@diku.dk

ABSTRACT

This paper gives an overview of research on the characterization of sounds by verbal attributes. Some of the difficulty of this characterization are highlighted, and a model of musical sound is proposed that could improve the understanding of the characterization. This model, which is based on a signal based synthesis model with relevance to both perceptual and physical issues, is called the timbre model. Verbal opposition pairs are proposed for many of the timbre model parameters (the timbre attributes). The combination of perceptual and physical relevance and verbal attributes are believed to be useful in the cognitive processes, such as memorization and recognition of synthetic sounds.

1. INTRODUCTION

It is important to be able to talk about the sounds. This can be done using verbal oppositions. A piano sound is dull, hard, clean, calm and decaying, while a viola maybe is bright, soft, dirty, disturbed and sustained.

Other, just as important verbal attributes, that describe the expression, the way the instrument is played, are equally important. It is possible to create both the timbre and expression of musical sound, using these attributes. This paper focuses on the characterization of musical sound using verbal attributes.

Most people probably understand if the violin in a piece of music is too hard, or the guitar is too dull, but what does it mean, if the flute is too wild? And what to do, if the listener of the music asks for a wilder flute?

The outcome of research on the verbal attributes of musical sound may be very useful in finding common terms for objectively describing music and music sounds. This paper presents a short overview of research in this area, and proposes to use the timbre model [4] parameters, the timbre attributes, as the basis for the verbal characterization research.

2. VERBAL ATTRIBUTES OVERVIEW

Timbre is best defined in the human community outside the scientific sphere by its verbal attributes (historically, up to and including today, by the name of the instrument that has produced the sound). von Bismarck [1] had subjects rate speech, musical sounds and artificial sounds on 30 verbal attributes. He then did a multidimensional scaling on the result, and found 4 axes, the first associated with the verbal attribute pair dull-sharp, the second compact-scattered, the third full-empty and the fourth colorful-colorless. The dull-sharp axis was further found to be determined by the frequency position of the overall energy concentration of the spectrum. The compact-scattered axis was determined by the tone/noise character of the sound. The other two axes were not attributed to any specific quality.

Rioux [2] did a thorough work on the verbal description of organ sounds, dividing the characterization into general, steady state and transient. He further found simplicity, tension, clarity and weakness, darkness and strength, complexity and appreciation for the first, leaking, sharp, fluctuation, clarity, rough and tension for the second, and duration, aggressiveness, softness and strength for the third. These prototype categories were further divided into 85 descriptors.

Sandell [3] collected 204 words from different orchestration books to characterize orchestration timbre. Verbal attributes have also been used to characterize sound quality in many studies.

Generally, most verbal attributes used in timbre research have been found using synaesthetics, i.e. these attributes were imported from other modalities. For instance, low-high in physics. Therefore, multi-modal confusion can occur if the modality is not specified. Other words are onomatopoetic (imitative) or physical.

One of the goals of a good verbal attribution scheme is to have good correlation to perception, physics or signal measures. In addition, a bipolar scale (verbal opposition pairs) ensures qualification, which permits quantification, for instance low-high (pitch), which is a continuous scale often discretized into note and octave. This (discrete) quantification seems necessary in order to make a categorization, which is probably an important step in the cognitive process. Hence, use of verbal attributes would enhance the memory, use and recognition of timbre with non-physical origin.

3. TIMBRE ATTRIBUTE VERBALIZATION

In this part the signal model which is used for the verbal attribution pairs is detailed, with the pertinent parameters, along with the corresponding proposed verbal opposition pairs.

3.1 Synthesis

The choice of underlying model in this work is the additive (sinusoidal) model, for its well-understood parameters (time, amplitude and frequency) and for its proven analysis methods.

The additive analysis consists in associating a number of sinusoid with a sound, and estimating the time-varying amplitudes, $a_k(t)$, and frequencies, $f_k(t)$, of the N sinusoids (partials) from the sound. The sound can then be resynthesized, with a high degree of realism, by summing the sinusoids,

$$s(t) = \sum_{k=1}^{N} a_k(t) \cdot \sin(2\pi \int_{\tau=0}^{t} f_k(\tau) / s_r d\tau). \qquad (1)$$

The additive parameters are further modeled by a high level model, the Timbre Model [4]. This model has been created to relate the perception with acoustic parameters. It consists of a spectral envelope, a frequency envelope, envelope parameters and irregularity parameters.

The amplitude envelope is modeled as a deterministic part,

$$\tilde{a}_k(t) = \hat{a}_k \sum_{s=1}^{5} \left(a_{s,0} + (a_{s,T} - a_{s,0}) e^{b_s (t-T_s)} \right), \quad (2)$$

where $\hat{a}_k$ is the spectral envelope value for partial k, s is the segment number (attack, decay/sustain or release), a_0 and a_T are the split-point relative amplitudes, b_s is the segment slope and T_s is segment length. Each segment is non-zero from T_{s-1} to T_s only. In addition, the amplitudes of the segments have irregularity introduced, called shimmer,

$$a_k = \tilde{a}_k(t)(1 + \sigma_k^a (c_k^a r_{0,BW}^a(t) + (1 - c_k^a) r_{k,BW}^a(t)), \quad (3)$$

where σ_k is irregularity standard deviation, c_k is correlation coefficient, and r_k is low-pass filtered gaussian noise with bandwidth BW. The same things apply to the frequencies, although the deterministic parts are assumed static,

$$f_k = \hat{f}_k(1 + \sigma_k^f (c_k^f r_{0,BW}^f(t) + (1 - c_k^f) r_{k,BW}^f(t)). \quad (4)$$

The frequency irregularity is called jitter.

The spectral envelope and frequency values can further be modeled as

$$\hat{a}_k = \hat{a}_0 \left(B / (B-1) \right)^{-k}, \quad (5)$$

which is the brightness creation function [4], $\hat{a}_0$ is the gain, and B is the estimated spectral centroid (brightness) [5], and the equation for the ideal stiff string for the frequencies [6],

$$f_k = k f_0 \sqrt{1 + \beta k^2} \quad (6)$$

where f_0 is the fundamental frequency and β is the inharmonicity coefficient.

In addition, the sound can be added tremolo (periodic amplitude modulation) and vibrato (periodic frequency modulation).

The synthesis can now be done using eqs. 1-6.

3.2 Verbal Attribution

In an attempt to enable the discussion of the timbre attributes, the two main attributes of the spectral envelope, the gain and the spectral centroid (brightness) are placed on the verbal opposition pairs *weak-strong and dark-bright* respectively, as seems to be the current use today.

The fundamental is defined here (and generally acknowledged) to correspond to the verbal opposition pair *low-high*. The inharmonicity is perhaps not a continuous scale, going from *straight* through some phasing to *bell-like*. The proposed opposition pair would be *straight-bell-like*.

The attack time is associated with the verbal opposition pair *hard-soft*, and the delay slope is associated with *sustained-percussive*.

The irregularities can give a large variety of modifications to the deterministic part of the sound, including additive filtered noise, jitter, agitation, etc., depending on the values of the standard deviation, bandwidth and correlation. Shimmer introduces additive noise for high bandwidths, and with something resembling a brass-quality for low bandwidth. The same effects occur with high correlation, but the sounds have a more *disturbed* quality. The jitter also adds noise for high bandwidths, but with a compact quality, and transforming into low-frequency jitter modulation for low bandwidth. In the jitter case, high correlation seems to give a more *synchronous* quality. These observations are not necessarily the same for sounds with different default attribute values, such as pitch, brightness, etc. In addition, the shimmer and jitter is multidimensional, with sound close to the origin labeled *organic*, *continuous*, *noiseless*, *resting*, *warm*, *lively* and *uniform*. With low correlation, the shimmer bandwidth increase is associated with, for instance, *metallic-noisy*, and with high bandwidth *interrupted-cracking*. The jitter bandwidth could be associated with *twangy-scattered* for low correlations, and *jittery-nervous* for high correlations. Other possibly attributions include *dirty*, *agitated*, etc.

Vibrato strength is temptingly associated with *still-swingy*, and the rate with *slow-fast*. The tremolo strength is associated with, for instance, *still-fluctuating*, and the rate with *anticipating-starting*.

4. CONCLUSIONS

This paper has given an overview of a synthesis model for musical sounds and presented a proposal for how the parameters of the model can be characterized with verbal opposition pairs. Some of these verbal attributions are believed to be in general use today, although further research is necessary to determine the validity of many of the proposed verbal attributes. In particular, the issues of synonyms and validity should be investigated, and on a larger scope, the cognitive issues should be better understood. Does the verbalization helps the memorization process, does it improve the mental organization, such that unfamiliar sound are more easily recognized, or even synthesized?

5. REFERENCES

[1] Bismarck, G. von, *Timbre of steady sounds*. Acustica, Vol. 30, 1974.

[2] Rioux, V., *Sound Quality of Flue Organ Pipes. An interdisciplinary Study on the Art of Voicing*. Ph.D. thesis, Report F 01-02, Chalmers University of Technology, Göteborg, 2001.

[3] Sandell, G. "*Verbal Descriptors from Orchestration Books*", <http://sparky.ls.luc.edu/sandell/sharc/verbal.html>, 27/09 2002.

[4] Jensen, K. *Timbre Models of Musical Sounds*, PhD. Dissertation, DIKU Report 99/7, 1999.

[5] Beauchamp, J. *Synthesis by spectral amplitude and "Brightness" matching of analyzed musical instrument tones*. J. Acoust. Eng. Soc., Vol. 30, No. 6. 1982.

[6] Fletcher, H. *Normal vibrating modes of a stiff piano string*, J. Acoust. Soc. Am., Vol. 36, No. 1, 1964.

Computers: From Calculation to Culture – The HCI landscape in a Historical Perspective

Anker Helms Jørgensen
The IT-University of Copenhagen
Glentevej 67, DK-2400 Copenhagen NV, Denmark
anker@it-c.dk

ABSTRACT

The computer has undergone tremendous changes in the last 50 years: from a calculation engine to a social agent. The computer is now a universal and pervasive device that must be viewed as a cultural and social phenomenon in order to be understood and exploited. Initially the computer belonged to the technical realm, used by indidual specialists, but along with the development a number of novel academic disciplines emerged (e.g., HCI and PD), drawing upon an even greater number of basic academic disciplines (e.g., Sociology and Art History). This paper presents a personal, historical overview of these developments in accompanying academic disciplines as seen from a HCI perspective. The overview takes shape as a table with seven entries: applications of computers; techniques; paradigms; interaction forms; forms of HCI knowledge; novel emerging disciplines; and basic disciplines.

Keywords
Computers, development, history, HCI perspective.

1. INTRODUCTION

In 2001 I had a very rewarding experience: I came across (and read) Steven Johnson's book *Interface Culture* [1]. Alas, the topic I had been working with for more than three decades had finally become a cultural phenomenon! The book gives a splendid account of the more recent developments of the computer, centered around the user interface, as seen from a cultural perspective. The publication of this book witnesses the encredible changes that have taken place in the last 50 years with the computer: from a calculator as used by engineers and astronomers to a social agent being used by millions of ordinary people in everyday activities.

This paper presents a personal view of the HCI landscape in a historical perspective. The paper aims in part to support newcomers in the field to get a grasp the origins of HCI and in part to provide grounds for a discussion of the field of usability that is being challenged by the social and cultural developments [2]. The exposition is by no means an attempt to tell the complete history of computers – this has been attempted by others, e.g., [3].

The landscape takes shape as a table on the following page. The table has seven entries:

- *applications of computers*
  where were computers being used primarily, e.g. in communication
- *techniques*
  what technical developments appeared that enabled the cultural and social penetration, e.g., networks
- *paradigms*
  how were computers seen in a general perspective, e.g., a tool for individuals in a work setting
- *interaction forms*
  what forms of interaction appeared, e.g., commands
- *forms of HCI evidence*
  what kind of evidence crystallized over the years, e.g., guidelines
- *novel emerging disciplines*
  which novel academic disciplines emerged over time, e.g., HCI
- *basic disciplines*
  which kinds of academic disciplines were drawn upon in understanding and exploiting computers, e.g. sociology.

2. REFERENCES

[1] Johnson, Steven (1997): Interface Culture. Basic Books.

[2] Jørgensen, A.H. (2000): The Concept of Usability: Challenges from Mobility and the Internet Proc. Work With Display Units, Berchtesgarden, Germany, May 22-25th, 2002, pp. 131-132.

[3] Wurster, C. (2002): Computers: An Illustrated History. Benedikt Taschen Verlag.

A Personal, Historical Overview of the HCI Landscape

Applications	Technique	Paradigms	Interaction forms	Forms of HCI Evidence	Novel Disciplines	Basic Disciplines
Calculations Administration Everyday things Communication Education Entertainment Design Art Culture	Pioneers Mainframes Minicom-puters Personal computers Networks VR Robots Ubicomp	Technical The Individual Organisational context Social Context - identity Societal Context - democrcy - economy	Knobs & dials Commands Menus & forms Graphical User Interfaces Speech gen/synth Multimedia Gestures Physiology Touch	Experience Guidelines Cognitive Experiments Concepts Theories Development Methods Standards Evaluation Methods	Computer Science Cognitive Ergonomics HCI CSCW PD CSCL CMC	Electronics Ergonomics Cognitive Psychology Linguistics Semiotics Philosophy Pedagogics Communication Media Studies Organisational Theory Design Sociology Etnography Cultural Studies Aesthetics Litterary Theory Art History Drama Occupational Health

Evaluating the Usability of Mobile Systems: Exploring Different Laboratory Approaches

Jesper Kjeldskov
Department of Computer Science
Aalborg University, Denmark
Fredrik Bajers Vej 7
DK-9220 Aalborg Øst
jesper@cs.auc.dk

Mikael B. Skov
Department of Computer Science
Aalborg University, Denmark
Fredrik Bajers Vej 7
DK-9220 Aalborg Øst
dubois@cs.auc.dk

ABSTRACT
Evaluating the usability of mobile systems constitutes a potential challenge since e.g. mobile systems are typically closely related to activities in their physical surroundings. In our proposed research, we explore different approaches to evaluating the usability of mobile systems in different laboratory settings.

General Terms
Human Factors.

Keywords
Evaluations, usability, mobile systems, safety-critical, think-aloud.

1. INTRODUCTION
Usability evaluations have proven to be invaluable tools for assessing computerized systems according to user mistakes and satisfaction [1, 2]. However, evaluating the usability of mobile systems constitutes a potential challenge since mobile systems are typically closely related to activities in their physical surroundings and requires a high level of domain-specific knowledge [3]. This can be difficult to capture in expert evaluations or recreate realistically in a usability laboratory. Thus, moving the usability evaluation into the real world may seem like an appealing approach. However, conducting mobile device usability evaluations in the field raise a number of potential problems. First, it is difficult to study certain mobile collaborative systems in the field e.g. since the use of system is temporally and spatially distributed among several actors and furthermore these actors move around while using the system. This complicates setting up a realistic usability evaluation. Secondly, some mobile collaborative systems deal with safety-critical issues involving risks for people and equipment. This further prohibits exploratory evaluations of the system since mistakes cannot be tolerated. Finally, field evaluations limit the means of control and complicate the data collection. Hence, we need to explore evaluations of mobile systems in laboratory settings.

2. METHOD
Our on-going research explores the above outlined challenges of usability evaluations in laboratory settings by comparing different kinds of experiments where we try to evaluate the usability of highly specialized mobile system designed for professional users performing safety critical collaborative work task on board very large container vessels involving people on the bridge and on the deck of the vessel [4]. The three experiments vary on the type of test subjects and the level of realism in the laboratory setting. The first experiment is a laboratory evaluation with non-domain subjects where 6 subjects with no knowledge of the domain evaluate the system in think-aloud sessions in a traditional usability laboratory. The second experiment is a laboratory evaluation with domain subjects where 6 test subjects with knowledge of the domain evaluate the system in a think-aloud sessions in a traditional usability laboratory. Finally, the third evaluation is an advanced laboratory evaluation with domain subjects where 6 subjects with knowledge of the domain evaluate the system in think-aloud sessions in an advanced ship simulator. In this sense, we attempt to vary the level of expertise of the test subjects and the level of realism of the use situation. All evaluations are recorded on video.

The data analysis aimed at creating three lists of usability problems identified in each of the three evaluations. The video examination resulted in a list of usability problems describing each problem as experienced by the test subject.

3. TENTATIVE RESULTS
We started to analyze the results of our studies. Totally, 58 unique usability problems were identified in the three evaluations. 7 of the problems were identified in all three evaluations and on both the bridge and on the deck. Some of these problems relate interaction issues, e.g. nearly all test subjects had problems related which elements to interact with on the screen. Another universal problem was that many test subjects did not see all relevant state changes in the system.

Karat et. al. have previously explored the similarities and differences of applying different laboratory approaches to usability evaluations [5]. They found that user-based evaluations generate significantly more (and more relevant) usability problems that theoretical evaluations. Henderson et al have made an analogous examination that compares four different user-based evaluation methods and found that the usability testing with think-aloud generates most usability problems [6]. Both studies, however, concludes that usability testing with think-aloud is the most expensive in terms of time consumption [5, 6]. In our study, we cannot identify any significant differences in the number of identified usability problems in the three evaluations. In this sense, it seems that inexperienced non-domain test subjects find as many usability problems as experienced domain test subjects. Furthermore, the environment for the evaluation seems to have little or no impact on the number of identified usability problems. In the standard laboratory setting, 37 usability problems were identified in the first evaluation and 40 usability problems were identified in the second evaluation whereas 36 usability problems were identified in the advanced laboratory. However, our study seems to indicate that the advanced laboratory facilitates the identification of a higher number of unique problems. In summary, our study indicates that the quantitative measures of the evaluations do not bring any significant differences between the three types of evaluations.

Starting to analyze the qualitative results of our study we encounter a different outcome. The distribution of the usability problems is somewhat different for the three evaluations. All three evaluations result in the identification of unique problems for that particular evaluation. E.g. the non-domain test subjects experienced problems in understanding the correct order of commands. This was no problem to the domain subjects. If we look at the unique problems for the evaluations in the standard or advanced laboratories with domain subjects, we can identify a number of interesting issues. First, more of the domain subjects needed to cancel already issued commands. This was not possible in the tested system. This turned out to be a critical problem in the evaluation in the advanced laboratory since the captain in the bridge had to apply different means of communication in order to cancel the command. This further resulted in problems of issuing new commands in the system. None of the non-domain subjects wanted to cancel commands or issued that this could be a problem. Secondly, the realism of the environment of the advanced laboratory resulted in that the test subjects (especially on the bridge) had to operate and consider other information resource that provided by the investigated system, e.g. the captain had to operate and navigate the actual controls of the vessel. Hence, the users' attention on the system was often disturbed that resulted in the fact that test subjects often missed updates or state changes in the investigated system. This was no significant problem to users in the standard laboratory. Thirdly, the complexity of the task imposed by the advanced laboratory, e.g. the conditions of the harbor in terms of other ships etc, made an impact on the results of the study. Some of the test subjects on the bridge wanted to apply a different approach to the command of letting go the lines. The system was restricted in that operation where certain sequential procedures could not be avoided. However, the conditions imposed by the advanced laboratories made the test subjects request different procedures in the operation. None of the test subjects in the standard laboratory evaluations discovered that problem.

4. ACKNOWLEDGMENTS

The authors would like to thank all test subjects and Svendborg International Maritime Academy for access to their ship simulator. The prototype was developed in cooperation with Jan Stage and Center for Human-Machine Interaction, from which we thank Peter Bøgh Andersen, Morten Lind, Morten Nielsen, Thomas Koester and Michael May for constructive comments.

5. REFERENCES

[1] Molich, R. (2000) *Usable Web Design* (In Danish). Ingeniøren|bøger

[2] Nielsen, J. (2000) *Designing Web Usability*. New Riders Publishing

[3] Nielsen, C. (1998) Testing in the Field. Proceedings of the third Asia Pacific Computer Human Interaction Conference. Werner, B. (ed.), *IEEE Computer* Society, California, pp. 285-290

[4] Kjeldskov, J. and Stage, J. Designing the User Interface of a Handheld Device for Communication in a High-Risk Environment. In Proceedings of the 7th ERCIM Workshop on User Interfaces for All, 2002, Paris, France

[5] Karat, C. M., Campbell, R., and Fiegel, T. (1992) Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation. Proceedings of the ACM CHI92 Conference. Reading, MA: Addison-Wesley, pp. 397 – 404

[6] Henderson, R., Podd, J., Smith, M., and Varela-Alvarez, H. (1995) An Examination of Four User-Based Software Evaluation Methods. Interacting with Computers, vol. 7, no. 4, pp. 412-432

Interaction as Negotiation

Jannie Friis Kristensen
Department of Computer Science
University of Aarhus
Denmark
+45 8942 5632
jannie@daimi.au.dk

Christina Nielsen
Department of Computer Science
University of Aarhus
Denmark
+45 8942 5659
sorsha@daimi.au.dk

ABSTRACT
Our recent work is focusing on presenting a general Human-Computer Interaction approach to designing technology with focus on facilitating negotiation in and between webs of artifacts, humans and places. Inspired by extensive empirical studies of mobile and nomadic work as well as the visions of ubiquitous and pervasive computing, we are concerned with the way technology presents itself to us in interaction situations, both as physical entities and as conceptual entities, as well as the relations between these, as we move between different work settings and tasks. This focus incorporates a much-needed attention on availability, visibility and connectivity as fundamental for understanding and supporting Human-Computer Interaction in relation to nomadic work and pervasive computing. Finally we discuss general challenges for users, designers and software developers in relation to ubiquitous human-computer interaction.

Keywords
HCI, design, negotiation, pervasive computing, web of technology.

1. INTRODUCTION
Scenario: *Paula leaves her office to do a presentation in the neighboring building. She brings her laptop and a stack of overheads with her power point presentation – just in case. She sets up her computer on the presentation desk and glances at the bundle of cables collected on the desk. She quickly identifies the cable for the projector sitting just under the ceiling and hooks her computer up to it. Next to the cables, Paula finds the remote control for the projector and pushes the "on" button. Nothing happens. She pushes the button again, this time harder, but with the same result. Extending her arm as far as possible, she points towards the projector and pushes the button several times, still no response from the projector. Then she recognizes a tiny orange LED-lamp on the projector and tries the "on" button again – she notices the lamp turns green on the third push of the button and turns to the screen behind her which, however, looks as dead as before. She decides to give it some time seeing as she got some kind of reaction from the projector and sure enough: after a minute or two of warming up, the familiar brand name of the projector comes up on the screen....*

As illustrated in the scenario above, the central design issue of the present paper is not whether the person is mobile or stationary, neither the specific technologies mentioned or the specific way they support the work situation. The core focus is to understand and formulate HCI principles and methods needed to make sense of ubiquitous environments, a general aim to present optimal possibilities for doing what ever the person wants, where she is, with the possibilities environments and artifacts present at a given point in time, gives. In other words, the general aim is not to be able to work with "anything-anytime-anywhere", but rather to get to know "*what* is available-*where* you are- and *how* it can be utilised" in relation to solving a specific work-task in a specific work situation.

Our world is becoming increasingly inhabited by computer devices, and many of them are more or less recognisable as such, built into tables, pens, chairs, clothing etc. These devices as well as the more or less visible connections between them, are presented to us as part of the visions of pervasive- and ubiquitous computing, and they provide us with many new opportunities, but also new challenges in relation to interaction. It may be challenging enough to make use of technologies that are visibly and recognizably present in our environment, as depicted in the scenario, but it becomes even more challenging how to make use of resources we cannot see or even assess as useful, because they are embedded in environments and artifacts. Thus, our broader research goal is to support the development of fundamental principles of ubiquitous human-computer interaction, and subsequently aid in presenting new design- and evaluation-methods in relation to the development of ubiquitous computing environments.

2. INTERACTION AS NEGOTIATION
Our HCI design approach; "Interaction as Negotiation" is generally inspired by extensive empirical studies of nomadic workers. This work context is characterized by a combination of mobile and stationary work, and constantly new constellations of work tools, places and people. Work-tasks are often solved in transitions between other tasks, and under unforeseen circumstances, and there is an extensive amount of work involved in coordinating, combining, gathering and finding work relevant information. We have found that the nomadic work setting combined with the "Interaction as Negotiation" approach presents a range of new challenges and implications in relation to the design methods we utilize in a cooperative design process. "Interaction as Negotiation" is directed towards the complete work organisation and context of nomadic workers, focusing specifically on the consequences of constant transitions between different tasks, places, situations and tools. With this focus we aim at getting a better understanding of the formation of constantly changing webs of artifacts, humans and places that make up the dynamic context for designing meaningful human-computer interaction interfaces in ubiquitous computing environments.

In the context of our approach the concept "negotiation" refers to the mutual mediation and translation process, taking place in any HCI situation. We argue, that the interface-design and the general design rationale of the technological environments and artifacts, as well as the specific way these are implemented in our work environments, can either enhance or decrease the possibilities for successful interpretation of the other actors in the first place, and second for possibilities of negotiation of that interpretation.

2.1 Visibility, Availability, Connectivity

Supporting interpretation and negotiation brings specific attention to three aspects of design and mediation:

- Visibility
- Availability
- Connectivity

With the concept of "visibility" we wish to bring explicit focus on the way technology presents itself to us physically, and how use and interaction possibilities are visually perceived and acknowledged by the human actors. This focus could also include other "physical" presentation clues than visual such as sounds and smells, but we have not in current projects focused specifically on these presentation forms. Design issues relating to visibility show increasing importance when designing embedded technology, and when the design context is highly nomadic work settings.

With the concept of "availability" we bring focus on the conceptual understanding of the technological artifacts, as presented in the different use situations. Recognizing that a technology is present (e.g. physical visibility) is not in any case sufficient in order to be able to conceptually understand a) the possibilities the technologies represent b) how they relate to other entities present in the situation, and/or c) how to interact - let alone negotiate - with it.

Finally, with the focus on "connectivity" we wish to bring focus on possible inherent contradictions between simplicity of use on one side and embedded technology and seamless integration on the other. An important issue here is that in the design of new artifacts and user interfaces we cannot afford to focus on a "single device - single user - specific context analysis" – but have to take into account ad-hoc formations of larger webs of artifacts, humans and places.

The negotiation between users and technology is modified by changing 'levels of significance' related to our need for specifying visibility, availability and connectivity in relation to a current object of interest in a specific situation context: we do not need to know where the fuse box is until a plug blows and we do not necessarily need to understand how a car is constructed, mechanically, to be able to drive one. But in every day work, we constantly change focus, either because we encounter a breakdown or because the nature of the task we are engaged in changes and then the level of significance inr relation to interaction changes to reflect this. The level of significance describes a dynamic structure through which we can filter underlying levels of complexity and meditate the interaction as negotiation process.

3. CHALLENGES IN UBIQUITOUS INTERACTION

Human-computer interaction in ubiquitous computing environments presents new challenges for users, and viewing this interaction process as negotiation presents interface and interaction designers as well as software architects with a new range of challenges.

Users in pervasive environments need an understanding of the opportunities/possibilities provided in them. They need to be able to recognize the presence of technology, and whether this technology is useful to them in the current situation. They also need an understanding of the relationships between the different types of technology. When the connections become less explicit, we need to compensate for it in the user interface, to support both use and recovery from breakdowns.

Interaction and interface designers of information technology are faced with the challenge of supporting the use aspects mentioned above: the design must reflect the need for making possibilities visible but not obtrusive; connections explicit but only when needed; and create flexible interfaces plastic enough to allow for the building and dissolving of ad hoc networks but maintaining a conceptual structure that is stable and recognizable enough for users to work within without getting lost. This in turn presents new challenges in relation to user-oriented design methods and for the methods and tools we use to evaluate and asses the usefulness of new technologies and tools.

Software architects in pervasive computing face challenges of creating technical building blocks, e.g. software infrastructures to build upon, for realising the vision of ubiquitous computing environments. This calls for the development of standards for information transfer or the building of flexible platforms that will allow heterogeneous devices to work seamlessly together.

4. SUMMARY

The area of human-computer interaction is changing to reflect the advances in technology and use practice within pervasive computing environments. As a research area we need to create a general foundation for ubiquitous interaction, and with it new design and evaluation paradigms. We see *interaction as negotiation* as a general design approach to support nomadic work in pervasive computing environments and as a first step in forming general principles of ubiquitous interaction.

Eight Fallacies of Distributed User Interfaces

[Extended Abstract]

Jakob Eg Larsen
Technical University of Denmark
Elektrovej, Building 371
DK-2800 Kgs. Lyngby, Denmark
jlarsen@cti.dtu.dk

Michael G. Rose
Technical University of Denmark
Elektrovej, Building 371
DK-2800 Kgs. Lyngby, Denmark
rose@tele.dtu.dk

ABSTRACT

Communications networks are a fundamental component in distributed interactive systems. The implications of networks for distributed user interfaces are discussed based on *Deutsch's Eight Fallacies of Distributed Computing.* We argue that the fallacies are useful as a set of design and evaluation heuristics for distributed interactive systems.

Keywords

Interactive systems, user interface design, evaluation, communications networks, distributed systems

1. INTRODUCTION

Distributed computing environments have become the norm. Mobile devices with more dynamic connectivity require user interfaces to function in more complex environments than designers have traditionally had to deal with. Computer scientists have been dealing with fundamental technical problems of distributed systems for decades, investigating the differences between local and distributed applications [11]. The key differences between designing a distributed and a local system has been summarized as the Eight Fallacies of Distributed Computing [1].

HCI researchers have recently begun to research information appliances [7], e.g. interface design for small displays and changing contexts of mobile devices. The implications of networks on usability in distributed systems and appliances have only recently begun to be addressed [4]. In HCI the effects of the network have been treated in a largely ad hoc manner. As the network will be a central feature in future computing environments, it is necessary to investigate its effects in a consistent and comprehensive way.

Deutsch [1] [5] provides a heuristic guide to these issues in the form of a set of fallacies, which arise from attempting to treat a distributed application as a local application. The "Eight Fallacies of Distributed Computing" are: 1) *The network is reliable*, 2) *Latency is zero*, 3) *Bandwidth is infinite*, 4) *The network is secure*, 5) *Topology doesn't change*, 6) *There is one administrator*, 7) *Transport cost is zero*, and 8) *The network is homogeneous.* The fallacies can be seen as design heuristics, that is, a set of design principles that should be followed by all designers of distributed systems. A system built upon any of these false assumptions will not be resilient to the failures that are inevitable on a network.

2. DISTRIBUTED USER INTERFACES

The fallacies will have implications for the design and evaluation of user interfaces for distributed systems. As networks are inherently *unreliable* it is problematic trying to make it transparent at the user interface level that there is a network involved in the system. Network failures that cannot be dealt with occur, so the user interface must communicate unrecoverable failure situations to the user so that he knows the state of the system, and thus the reason why it does not fulfill requests.

Latency, which manifests itself in system response time, is critical in both local and networked applications. Dix introduces the fallacy of "The Infinite Fast Machine" [2], which is the assumption that machine response time is zero. However, as network latency tends to be orders of magnitudes longer it is even more critical in networked systems. Informing the user of progress is important in interactive systems, especially when the time taking to respond to an action is longer than 100–200 milliseconds [6] for the user to experience a continuous interaction. In many cases the distributed interactive systems will not be able to respond in that timeframe why the system must immediately indicate that the user's action has been intercepted and is being processed.

Bandwidth issues are also manifested through response times, e.g. long upload and download times as well as poor quality of service for audio and visual media. Indications of bandwidth must be provided to users when they are about to initiate a bandwidth intensive task, provide them feedback on the progress of the task, and provide lower bandwidth alternative for the task. Many of the user interface implications can be attributed to response times. Long response times are problematic, but if good progress indications are provided varying response times are worse [3]. That is the case in loosely coupled networks, such as the Internet and wireless networks. As users form habits from working repeatedly with the user interface [8], they will want to "set their own pace of interaction" [9].

Insecure protocols and applications such as e-mail and file transfer have historically been the default on the Internet.

These applications send data as plain non-encrypted text. The user interfaces of these applications generally do not indicate that they are insecure, or which level of security (if any) the user should expect. Security is crucial for the types of services that networks, such as the Internet, are being used for today.
Network *topology* is not often an issue for end users. However technologies such as peer-to-peer and spontaneous networking systems [10] allow the topology to change dynamically, and in the process drastically changing the services and information available to the user. Contrary to local systems there is no "current state" of distributed systems due to its dynamic features.
In a peer-to-peer system each participating peer could be considered an *administrator* of his own participating node. There is no guarantee that a peer administrator does not remove the node that contains the information that a user was searching for.
Different subscription channels to the network have different price structures and network bandwidths accordingly. The *cost* of network transport vary significantly dependent on the situation in which it is used. The user should have options that allow them to consider the additional cost and allowing them to skip or postpone retrieval of large content.
The network is not necessarily *homogeneous* as it can be made up of different types of networks with gateways between them. Special networks typically have specialized devices, e.g. mobile phones and networks. The information appliances, applications, and user interfaces used in one network is not necessarily fully adaptable to or interoperable with the information appliances, applications and user interfaces of the other network.

3. EXAMPLE: WEB BROWSERS

Web browsers are typical end-user distributed computing applications. The Mozilla web browser[1] provides a useful, if incomplete, example of the relevance of the fallacies for user interface design. The Mozilla design has an explicit *status bar* in the bottom of the application window that provides feedback to the user about the state of the system. The online/offline mode indicates network availability. Network related failures are indicated through error messages presented in popup dialogues. An animated icon provides elementary feedback on network latency.
The status bar has a progress bar that informs the user of progress of downloading a webpage. Web browsers cannot display a *time* accurate progress bar to the user when downloading a page, but if possible indications of the total amount of data to be downloaded and the amount downloaded so far is presented.
A *lock* icon in the status bar indicates whether or not a secure transfer is used. In addition a warning is provided when the user switches from insecure to secure transfer mode, and vice versa.
The multiple administrator aspect is not dealt with explicitly. An example that illustrates this issue is the multiple registrations that the user has to keep in order to use different services on the World Wide Web. The user has to remember and provide multiple usernames and passwords for the various websites. Technologies addressing this problem are being developed, to provide a uniform user interface that handles the login mechanism for networked delivered services. The issue of transport cost is not dealt with in current web browser implementations. This is especially an issue for mobile access devices, where the cost vary dependant on location and access point.
The homogeneity issue is also relevant for web browsers. Many standard webpages cannot be rendered so they are readable on TV screens due to a lower display resolution compared to standard computer monitors. Even harder is bringing webpages to mobile terminals with small displays and low display resolution.

[1] http://www.mozilla.org/projects/ui/communicator/

4. CONCLUSION

The differences between local and distributed computing are fundamental and lead to design invariants in distributed interactive systems. We conclude that Deutsch's eight fallacies of distributed computing can be a useful heuristic guide to user interface designers and evaluators of distributed interactive systems.

5. ACKNOWLEDGMENTS

Thanks to Peter Deutsch for answering questions about the fallacies, and to Alan Dix for a draft of chapter 16 of [4].

6. REFERENCES

[1] L. P. Deutsch. Personal communication. Deutsch created the list of the "Eight Fallacies of Distributed Computing" when he was working at Sun Microsystems 1991–93. 2001.

[2] A. J. Dix. The Myth of the Infinite Fast Machine. *People and Computers III – Proceedings of HCI '87. Eds. D. Diaper and R. Winder*, 2(1):215–228, 1987.

[3] A. J. Dix. Cooperation without (reliable) Communication: Interfaces for Mobile Applications. *Distributed Systems Engineering*, 2(3):171–181, 1995.

[4] A. J. Dix. *Network-Based Interaction. Chapter 16 of Handbook for Human-Computer Interaction. Eds. A. Sears and J. Jacko*. Lawrence Erlbaum & Associates, in preparation.

[5] W. K. Edwards. *Core Jini*. Prentice Hall, 1999.

[6] R. B. Miller. Response time in man-computer conversational transactions. *Proceeding AFIPS Fall Joint Computer Conference*, 33(1):267–277, 1968.

[7] D. A. Norman. *The Invisible Computer*. The MIT Press, 1998.

[8] J. Raskin. *The Humane Interface*. Addison-Wesley, 2000.

[9] B. Shneiderman. Response time and display rate in human performance with computers. *ACM computing surveys*, 16(3):265–286, 1984.

[10] J. Waldo. The Jini architecture for network-centric computing. *Comm. of the ACM*, 42(7):76–82, 1999.

[11] J. Waldo, G. Wyant, A. Wollrath, and S. Kendall. A note on distributed computing. Techreport TR-94-29, Sun Microsystems Laboratories, 1994.

Widget multiplexers for in-situ handling of alternative application states

Aran Lunzer
aran@bigfoot.com*

Abstract

The Subjunctive Interface approach, introduced in previous work, assists users of interactive applications in exploring the relationship between application inputs and results. The key to the approach is presenting alternative values simultaneously, side by side, for ease of comparison. For some styles of input and result this can be achieved simply by visual merging or overlay of the alternatives, but for other types of information, such as text or dense graphics, this is clearly unsuitable. Furthermore, restrictions on layout and space militate against duplicating entire displays. The techniques described here come to the rescue, being based on coordinated 'widget multiplexers' that condense multiple versions of display regions to enable side-by-side comparison without excessive use of space.

1. Working with multiple versions of reality

Many activities involve human-guided explorations among a domain of computer-generated results, which in unpredictable domains often reduces to trial-and-error search. To increase search efficiency and thus reduce frustration, the *subjunctive interface* approach [1, 2] enables pursuit of many strands of enquiry in parallel. A user sets alternative provisional values for application parameters, thereby defining alternative states of the underlying application (referred to as alternative *realities*), then steers these realities in parallel through the search space. The results encountered in all the realities are displayed simultaneously, juxtaposed for ease of comparison.

Fundamental to the subjunctive interface approach is that the parameters to be set, and the results that are obtained, are handled within their original application context rather than being extracted and normalised to suit some generic multi-attribute visualisation technique. One motivation is that not all explorations are quick optimisations, finalised in a single session; some can involve long-lived investigation of alternatives. This is true for the prototype shown here: a demonstration of how a personal calendar tool might let its user schedule undecided events for multiple possible dates, then see how those alternatives work out as new commitments arise over the intervening weeks or months.

The main challenge for a subjunctive interface is presenting the multiple realities simultaneously, while helping the user keep track of which results correspond to which provisional settings. Early demonstrations addressed cases in which the realities could be juxtaposed by simple graphical overlay. The work shown here is a step towards a more general approach, suitable for applications where overlay is not a viable option.

2. Widget multiplexing

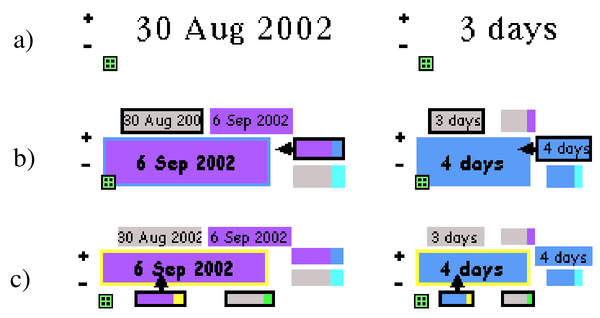
A widget multiplexer takes control of the display region occupied by some part of an application, and manages within that region the different appearances and behaviours it would have in the application's various realities.

This is a multiplexer operating on a ScheduleBlender calendar view. If the user were working with only one reality, the calendar would take up the entire region of this display. In this case there are three realities, whose views

*The author is currently on a self-funded sabbatical. However, much of the work presented here was conceived during his employment at the Meme Media Laboratory, Hokkaido University, Japan.

are visually compressed by scaling and pseudo-3D tilting and are arranged around a relatively large, flat *working view*. The working view replicates the contents of the *primary reality* as selected by the user (currently the one whose view is at top left), ensuring that one reality can be seen and manipulated at a comfortable scale.

Different styles of widget call for different styles of multiplexer. Limited screen resolution constrains the readability of tilted views, making them unsuitable for distortion-sensitive displays such as text strings. Therefore a specialised textual multiplexer has been developed, as shown in the following figure.



This shows two multiplexers dedicated to the start date and duration for some calendar event. Row (a) corresponds to having just a single reality, while (b) and (c) show four and six realities respectively.

Like the graphical multiplexer, the text version shows reduced (though now rectangular) views around a larger working view. Notice that the two multiplexers are coordinated to have the same arrangement of views (reality 1 always appears at top left, with the others arranged clockwise from there). This brings the familiar 'small multiples' advantage of helping users locate corresponding views in different contexts, and is therefore enforced even for multiplexers that do not have different values to show in every view. That said, displaying the same value in many views would cause visual clutter and make it harder to tell which values agree. To avoid this, the textual multiplexer only shows each different value in one of the views, then uses colour coordination to reveal other views sharing that value. In the right-hand part of row (c), for example, realities 1, 2 and 5 share the value '3 days', while 3, 4 and 6 are all '4 days'. In addition, when a multiplexer has the same value for all its realities, by default it reverts to a single full-sized view.

The dark outlines seen here around some of the views show reality selections, made by the user. The selection of primary reality is augmented by an arrow pointing towards the working view. An outline with no arrow indicates a subsidiary selection, turning that reality into a 'follower' of the primary. Any operation performed on the primary reality, such as incrementing the length of an event, or a meta-operation such as duplicating the reality to create a new one, is echoed to its current set of followers. Indeed, row (c) in the second figure is the outcome of a 'duplicate' operation on the state in row (b), where reality 3 was primary while reality 1 was selected as a follower.

The two examples above demonstrate multiplexing at different granularities – the former juxtaposing entire two-month calendar views, the latter just the settings for individual values. In general the user should be free to control this granularity, in accordance with the kind of comparison wanted. In support of this, widget multiplexers can be nested. For example, in ScheduleBlender the events' starts and durations appear in a list, and the list has its own multiplexer; the user can choose whether the multiplexing should take place at the level of the individual values, as seen above, or at the level of the entire list. Since the event list is sorted by the events' start dates, one advantage of comparison at the list level is that it becomes easy to notice if different realities involve different event orders.

3. Reality management

This description has introduced only the basic interaction concepts; any practical application of the technique must include provision for creating and managing the necessary range of realities. The current multiplexer designs accommodate up to eight realities – which in itself is challenging for a user to handle at one time, especially if all eight are changing in parallel – but some applications may simultaneously embody many times this number. One reason is the use of long-lived realities, such as the provisional dates mentioned here; another is if a user explicitly chooses to explore the combinatorial effects of provisional values for multiple parameters. This is the focus of ongoing work.

References

[1] Lunzer, A.: Towards the Subjunctive Interface: General Support for Parameter Exploration by Overlaying Alternative Application States. In Late Breaking Hot Topics Proceedings of IEEE Visualization '98, Research Triangle Park, NC, USA (1998) 45–48

[2] Lunzer, A. Choice And Comparison Where The User Wants Them: Subjunctive Interfaces For Computer-Supported Exploration. *Proceedings of 7th IFIP Conference on Human-Computer Interaction (INTERACT '99)*, Edinburgh, Scotland, Aug/Sept 1999, 474–482.

Time and Learning in HCI

Marianne Graves Petersen
Department of Computer Science
University of Aarhus
Aabogade 34, 8200 Aarhus
mgraves@daimi.au.dk

ABSTRACT
Most usability evaluations provide snapshots of use as the basis for recommendations on re-design. This abstract suggests an alternative perspective, namely that of considering time and learning in HCI. This implies understanding use as dynamic and developing, and considering how users learn to use the technology over time. Introducing a perspective of time and learning in usability work has both theoretical and practical consequences as discussed in the following.

USABILITY EVALUATION AND LEARNING IN USE
Usability evaluations based on studying 30 minutes interaction with a prototype in a laboratory context misses out central issues, which we cannot afford to loose, especially not as interactive technology penetrates more and more aspects of our everyday-lives. New approaches are needed to allow for a focus on exploratory learning and on users' future competence and on how this can be supported by the design of everyday artefacts.

Insights from activity theory can help us understand this challenge. In considering learning within the framework of activity theory, Vygotsky (1978) emphasises the need to look beyond the present competence of users, and consider prospective development, through investigating the zone of proximal development. The zone of proximal development

"is the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" (ibid, p. 86).

Interpreting this perspective onto practical evaluation and design of interactive technology has a number of implications. For instance in the traditional set-up of usability testing, focus is exactly on the present competence of users and not on the prospective development, i.e. how the design of the technology creates a zone of proximal development. In traditional usability testing users are taken into a lab, given a scenario or use-case and are then asked to perform a task as described in the scenario (Borgholm and Madsen 1999). The focus here is then on whether the user can accomplish the task and where she has problems. Throughout the test, the test-leader should not interfere with the test (Tognazzini 1992) and has traditionally been placed in a separate room with one-way mirrors into the test-room (Buur and Bagger 1999).

Looking into the zone of proximal development in usability work requires a different approach than the currently predominant. In the following some examples of how this may be accomplished are presented.

We have developed an explorative inquiry approach Kjær et al. (2000), which attempts to create a set-up allowing us to look into how design supports the creation of a zone of proximal development. Instead of being isolated from users, we sit down with them in their natural context. Without revealing the functionality of the product, we ask questions, which focus on the new experiences people have had with the technology. We ask about the things they would like to try out in the future, which functionality and opportunities they expect from the product and how they have become aware of these opportunities. Further, in evaluations it is important to not only focus on users' actual competencies but to look towards potential compentecies, through understanding what people can do with a little help.

Considering pre-use
A second consequence of considering learning in use of artefacts is that expectations are often set even before people lay hands on the products. E.g. Petersen et al. (2002) illustrates how B&O customers have a rich set of expectations for the functionality of the new TV set, and how this affects the later adoption process. Thus evaluating learning in use implies a concern for considering pre-use and how or if users are motivated to getting started in the first place, since this is a big part of the trick in the case of everyday artefacts like B&O products (Bærentsen 2000). Thus interviews on expectations before use, preferably in a natural context, is an important part of understanding the potential usefulness of the technology.

The technology tour
The technology tour is developed with a particular concern for the characteristics for the household. While the term Technology Tour signals a focus on the technology itself, the idea of object biography is that it can act as a tracer of the changing social and cultural contexts of objects (Silverstone et al. 1992). It thereby provides a way to understand everyday artefacts in the broader context of their use activity, cf. discussion above. The Tour consists

of asking the householders to give a round of their house going through each technology and tell its story triggered by questions posed by the investigator. The idea of object biography is not elaborated much by Silverstone et al. (1992), since they have an interest in understanding technology as media, rather than as tools. Therefore we developed it as described in Petersen & Baillie (2001) for our interest in understanding present technology use, to provide a basis for developing new technology. On the tour we focus explicitly on

The role of the physical space (Silverstone et al. 1992, O'brien and Rodden 1997, Venkatesh 1996)

- how the technology is situated in the physical space, motivations for this organisation
- the stability of the physical organisation – is it changed in special situations
- possible conflicts of ownership of space
- the history of the physical organisation – how was it organised before, why was it changed
- satisfaction / dissatisfaction with the present organisation

Rules and ownership of space and technology (Engeström 1987, O'brien and Rodden 1997)

- Who has the right to do what and when
- Who has the right over whom and in what situations
- Possible conflicts, when and why

History of the technology (Engeström 1987, Bødker 1996, Silverstone et al. 1992)

- Who acquired it when and why
- Describe problematic situations you have had during times of use

Present use of the technology (Bødker and Grønbæk 1991)

- Create hands on situations based on scenarios constructed from the stories told by the members of the household
- What is favoured/disliked about it

Our experiences from the Technology Tour suggest it has the following characteristics

- It is a good start up activity. People find it quite easy to get started and you start to get a feeling of the home. At best, the whole family shows around, making discussions on possible conflicts and rules more easy going and interesting, providing a less idealised picture of use practices.
- It gives the householders the ownership of the situation, making them show the way around, and set the limits as to how far around they want to go. Some view the bedroom as a very private space. Others do not care so much.
- We got particularly interesting accounts in this way, when people started to tell stories (Brown & Duguid, 1996). The stories are records of particular incidents, and "they are built in response to the particulars of breakdown" (ibid, p. 65). For example John's story about the first time they used the stove. John and Susanne were about to cook dinner, but they could not work out how to turn on the stove. Puzzled by the complexity of this apparently simple operation, they finally gave up, called the previous owner of the flat, and got him to tell them how to turn on the oven. Subsequently, they have several times used this as a "party trick", asking people to turn on the stove, having a laugh as it turns out each time that people cannot accomplish this on their own. The stories are valuable in that they "are repositories of accumulated wisdom" (ibid, p. 66) or in Venkatesh and Mazumdar's (1999) terms they provide accounts of both the social, physical and technological space of the situation, thus providing holistic accounts of the event.

TIME AND LEARNING IN USABILITY DESIGN

As argued earlier on, time is an important factor when studying learning in use of everyday artefacts. Thus taking up the idea that cycles of use and design mirror each other (Engeström 1987 and Bødker 1999), we experimented with means of introducing time in the design process.

Living with mock-ups

One approach to do this, again directed towards domestic technology, is to ask people to live with mock-ups and investigate how they become appropriated over a period of time. In our experiment, we asked the families to concretise their ideas and visions into mock-ups. As in the technology tour we also focussed explicitly on the physical placement of the mock up, asking the users to discuss and place the mock up in the physical context of their home. We finally extracted scenarios from their descriptions.

We left the mock-ups with the families, in the place where they had themselves positioned them and we asked the families to live with them over a period of time. We provided them with post-its and encouraged them to further mould the mock ups, move them, add on any ideas, reflections, critiques etc. as they occurred to in the context of their daily life practice and family life.

The idea here is much in line with the use of mock-up in cooperative prototyping (Ehn and Kyng 1991), where systems developers and users cooperate on developing a new design through discussing and simulating use on mock ups, situated in the actual context of use. What is different here is that we ask the users to dwell on the mock up in the context over a prolonged period of time and to materialise their comments and ideas on post-it notes, as they occur to people over time. Our approach shares the idea of gathering information from households over time with the cultural probes idea by Gaver et al. (1999). Gaver et al. (ibid) provide people with different means of communication including disposable cameras and pre-stamped post cards asking provocative questions like "why do we have politicians". However, in their approach, the resulting technology design, is created detached from the context of use, only linked to the community through the post cards and cameras people send. Instead, we have sought to maintain a close collaboration with the users. A session from one of the families serves to illustrate the approach.

After Sue and John have taken us on the technology tour (described above) they are presented with a number of pictures e.g. “intelligent” microwave oven supporting banking etc, and a number of different technological devices.

Following up on earlier discussions Sue mentions the idea of having recipes on a screen in the kitchen. We look at and discuss the use of her own personal recipe book, which her mother got for her. It contains only recipes, which has been tried out and found good. They are written in by hand and personal adaptations are made in this process. We discuss the idea of an electronic version for a while, and Sue is asked to materialise how such a device could look and where in the kitchen it could be. Together, we construct a first, crude representation of recipe support in the kitchen. It contains a pen, which allows for annotations and corrections inspired by the process of cooking. We also experiment with the idea of a “flying” representation triggered by a balloon and with having several representations, for when Sue and John are both in the kitchen preparing each their part of the meal. We finally elaborate on the idea of a screen – like device, which Sue argues should be placed lying on the table right next to the stove, as this is where she needs to see the recipes. A number of ideas and discussions come up linking the support for recipes to a family calendar showing when guests are due and also support for a shopping list. The discussions also turned down a number of common ideas on support for the kitchen. Sue found the detailed account of the contents of the fridge uninteresting. Instead she would like to be able to “look into the fridge” while at work preparing the shopping list. When the electronic recipe book is used to support cooking, pictures are not very important. These are more interesting in a situation of recipe surfing, which typically happens as a relaxed situation in a soft chair. As the session finishes, the mock up is left on the table beside the stove, and Sue and John are provided with post-it notes and asked to note down ideas, comments etc. At the next visit, it turns out that the mock up has been moved to a different place in the kitchen. They found that as they start to cook in the kitchen, the place on the table is needed for preparing the food. Thus the mock-up has been moved to the wall next to the fridge. They also suggest that in this place, it serve mainly to guide the cooking of a specific meal. This is not the place to surf for recipes. Sue suggests that its appearance should be fancy and funny, in line with the Allessi products occupying their kitchen. She suggests the form of a silver heart.

CONCLUSIONS

This abstract makes the case for considering time and learning in usability work. It provides an example of a theoretical framework for understanding these aspects of use and it offers some practical techniques for doing so, some of them particular directed towards another of the challenges for HCI, namely that of domestic technologies. The ideas presented here are further deveveloped in Petersen (2002).

REFERENCES

Borgholm, T. & Madsen, K. H. (1999) Cooperative Usability Practices. In *Communications of the ACM*, vol. 42, No. 5, pp. 91-97.

Brown, J. S. & Duguid, (1996) Organizational Learning and Communities of Practice. Toward a Unified View of Working, Learning and Innovation. In Cohen, M. D. & Sproull, L. S. (Eds.) *Organizational learning.* London, Sage, pp. 58-82.

Buur, J. and Bagger, K. (1999) Replacing Usability Testing with User Dialogue. In *Communications of the ACM*, vol. 42, No. 5, pp. 63-66.

Bærentsen, K. B. (2000) Intuitive User Interfaces. In *Scandinavian Journal of Information Systems*, vol. 12, pp. 29-61.

Bødker, S. and Grønbæk, K. (1991) Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. In Greenbaum, J. & Kyng, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems*. Lawrence Erlbaum Associates, pp. 197-218.

Bødker, S. (1996) Applying activity theory to video analysis: How to make sense of video data in HCI. In Nardi, B. (Ed.) *Context and Conciousness. Activity Theory and Human Computer Interaction,* Cambridge: MIT Press, pp. 147-174.

Engeström, Y. (1987). *Learning by expanding*. Helsinki: Orienta-Konsultit.

Kjær, A., Madsen, K. H. & Petersen, M. G. (2000) Methodological challenges in the study of technology use at home. In Sloane, A. & van Rijn, F. (Eds.) *Home Informatics and Telematics: Information, Technology and Society,*. Boston: Klüwer Academic Publishers, pp. 45-60.

O’brien, J. & Rodden, T. (1997) Interactive Systems in Domestic Environments. In *DIS’97 Designing Interactive Systems: Processes, Practices, Methods & Techniques*, Amsterdam, The Netherlands, ACM., pp.247-259.

Petersen, M. G. and Baillie, L. (2001) Methodologies for Designing Future Household Technologies. In *Proceedings of the OIKOS2001 Workshop,* Aarhus University Press, pp. 47-49.

Petersen, M. G., Madsen, K. H. and Kjær, A. (2002) Usability of Everyday Technology – Emerging and Fading opportunities. In *ACM Transactions on Computer-Human Interaction*, Vol. 9, No. 2, June 2002, pp. 74-105.

Petersen, M. G. (2002) Designing for learning in use of everyday artefacts. Ph.D. Thesis, University of Aarhus.

Tognazzini, B. (1992) *Tog on Interface.* Apple Computers, Chapter 4, pp. 79-89.

Venkatesh, A. and Mazumdar, S. (1999) New Information Technologies in the Home: A Study of Uses, Impacts, and Design Strategies. EDRA Conference(Environmental Design and Research Association), Orlando, Florida, Jun 2-6, 1999.

Vygotsky, L. S. (1978) *Mnd in Society. The Development of Higher Psychological Processes.* Harvard University Press, Cambridge, Massachusetts.

Use of human centered stories for describing Human Computer Interaction

Georg Strom
Department of Computing Science,
University of Copenhagen
Universitetsparken 1
DK-2100 København Ø.
+45 40 18 12 72
georg@diku.dk

ABSTRACT

This paper describes a comparative study with six stories and six technical texts and a total of 15 participants from information technology organizations. The study shows the advantages of stories having an interesting plot and plausible characters. It demonstrates a large interest in using such stories when making decisions, that fictive stories can facilitate more detailed discussions about human-computer interaction and make decisive changes in the reader's attitudes to specific problems, even though the amount of factual knowledge gained from both stories and technical texts is low.

1. Introduction

In order to design the interaction between a human and a computer interface, it is necessary to describe it. A story can give a very concentrated description of the user's experience or interaction in a specific situation. [4].

I will investigate the preferences and value of human-centered stories, stories that describe the human computer interaction with a focus on human motivations and emotions: similar to the focus in most published short stories and novels.

2. Method

The investigation of the use of human-centered stories consists of the following parts:

- I investigated which aspects of the use of information technology that could be described within the limitations of different types of stories, in particular scenarios.
- I wrote six human-centered stories, short stories, where each story describes the use of a specific type of information technology.
- Four of the stories were read by specialists in the topics described in them, and I conducted structured interviews with one specialist in each of three areas and with two specialists in one area.
- In addition to the human-centered stories I wrote six technical descriptions covering the same topics. Ten mid-level managers, typical decisionmakers on requirements or acquisition of software, each read either a story or a technical text describing each of the six topics (each read three stories and three technical texts). I then conducted individual structured taped interviews with them.
- The processing of the data are still in progress and will be completed in the middle of October (2002).

3. Results

Erickson [1] writes that: ”Scenarios are often abstract – they are scripts of events that may leave out details of history, motivation and personality”.

I have reviewed 19 scenarios which either are used by companies or published [2] [3] [5]. All scenarios tell about the action more than show it, they have no dialogue, there is no other characterization of their characters' personalities or motivations, and the actions of the characters are often implausible.

Of the 19 scenarios, 16 can be characterized as daydreams with no human conflicts and where the interaction and the technology function without any problems.

In contrast to scenarios, short stories and novels are rarely read if they cannot entertain and enjoy the readers. That is only possible if they are human centered stories with engaging conflicts, and characters and a plot that within the context of the story is perceived as credible by the intended reader.

Short stories and novels have a sequence of events and a structure based on common conventions in story telling and a general knowledge about how people with different personalities react in different situations. This means that in a well-written story (in contrast to a scenario) it is possible to give a good characterization of a specific situation with only a few words.

3.1 Writing of human-centered stories

I participated in a creative writing course and three of the stories used in this investigation were reviewed in the course. In order to write readable human-centered stories, I found it is absolutely necessary to learn the basics of the craft.

Human-centered stories are only interesting if they include a certain amount of drama. I found that I while writing the stories tended to look for problems and conflicts in the human-computer interaction.

I found that the time it took to write one page of a human-centered story was about the same as the time it took to write one page of technical description.

3.2 Attitudes to stories

A number of managers stressed the need of literary quality and both domain experts and managers stressed that a dramatic element significantly increased the value of the stories. The entertainment value appeared to be a major argument for the use of stories for describing human-computer interactions.

In contrast, the managers perceived the stories with the most dramatic or exotic action as significantly less trustworthy (2.17 versus 2.97 on a scale with 1 as best and 5 as worst value). It is likely that the managers found that the less dramatic stories gave a better impression of the actual use of the described technology. (The number of domain experts was too small to make a statistic comparison.)

The managers rated that the stories compared to the technical texts gave a slightly better impression of the use of the described technology.

In 46 % of the cases, the managers said it would be natural for someone to refer to the story during a decisive discussion about requirements or acquisition of software. The domain experts reported a similar interest in the use of stories. One interview was done with two experts in the same domain, they gave the trustworthiness of the story (probably the most dramatic one) a low rating, but added that they would like to use the story to exemplify and explain certain problems to their management.

Three of the stories were written in first person, and three in third person (similar to the camera viewpoint used in scenarios). This difference in viewpoints had seemingly no influence on how the participants evaluated the stories.

3.3 Acquired knowledge from stories

The investigation indicates that the managers typically retained about the same amount of technical information from the technical texts as from the stories. However, the variation between the participants was significantly larger for the technical texts than for the stories.

The amount of retained technical information was in general low from both stories and technical texts, and the variation between the participants was much larger than the variation between the different types of text.

3.4 Changes in attitude after reading stories

In 87 % of the cases, the participants felt they could evaluate the usability of a software application after reading a technical text listing the features and functions available in it.

Two of the stories describes how the software is easy to operate, and two stories describes how the users have problems using the software, but it appears that these descriptions had only little influence on the perceived usability. However, compared to technical texts, stories gave a more varied view of the usability: When discussing a story several participants distinguished clearly between the usability for themselves, for the person in the story and for intended users in general. No such distinction was made when discussing a technical text.

Six specific issues or problems were demonstrated in length in the stories, for instance regarding work conditions or computer security. The results indicate that the stories changed the attitudes of the readers in a decisive manner on two of the issues; a smaller change was detected on three additional issues.

3.5 Role of stories in discussions

The domain experts gave more detailed replies after having read a story about a topic, and based on the story they were able to discuss precise details of the interaction and of the social aspects of the use of the technology. The managers who had read a story about a topic mentioned social aspects when discussing the use of the technology. Such aspects were only in one case mentioned after having read a technical text.

4. Conclusion and discussion

There is a significant risk of being misled when scenarios without any plausible characters are used to discuss needs and reactions of users and how the interaction will be between users and a specific interface. In addition, the participants stated that stories used for describing human-computer interaction should fulfill the same criteria's as normal short stories, for instance plausible characters. (A comparative study between current scenarios and human-centered stories may yield interesting results.)

In contrast to the writing of scenarios and technical descriptions, the writing of human-centered stories tends to highlight possible problems in the human-computer interaction.

The entertainment value is a significant motivation for using stories. However, stories are taken seriously and may be used as part of the basis for major decisions.

The results indicate that the best stores are realistic everyday dramas. The writers shall avoid major dramatic events both in general and in the lives of the characters.

Presenting technical information as part of a story may help those who are least capable of retaining information from a technical text. However, the amount of retained factual knowledge is low from both stories and technical texts. That may be a concern when people discuss technical details without having documentation available, as may be the willingness to evaluate usability based solely on a list of features and functions.

The investigation indicates that stories can change attitudes in a decisive manner, and the effect may be even stronger for people who have not followed any discussions of the specific issues (as most of the participants in the present investigation probably have).

Finally, the investigation shows that stories significantly facilitate more detailed discussions of the social context in which a technology is used, whereas texts with purely technical information tend to block such discussions.

5. REFERENCES

[1] Erickson, Thomas: Notes on Design Practice, ed. by John M. Carroll, John Wiley & Sons, 1995

[2] Harvey, Fiona: The Internet in Your Hands, in Scientific American, October 2000

[3] Niman, Bruno von, Mathias Schneider-Hufschmidt, Satu Ruusska-Kalliokulju, Kaisa Väänänen-Vainio-Mattila: The user experience of future mobile communication, HfT 2001 conference

[4] Norman, Donald A.: Things that make us smart, Addison-Wesley Publishing Company, USA, 1993

[5] Rosson, Mary Beth and John M. Carroll: Usability engineering, Morgan Kaufman Publishers, 2002