FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Preed – Hypertext Application for Sharing and Presenting Digital Artefacts in Meetings.

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik

eingereicht von

Stephan Bazalka
Matrikelnummer 0225725

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer/in: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Purgathofer

Wien, 13.03.2011 ______________________ ______________________

(Unterschrift Verfasser) (Unterschrift Betreuer)

Erklärung zur Verfassung der Arbeit

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, am 13. März 2011 Stephan Bazalka

Danksagungen

Diese Diplomarbeit wäre ohne die tatkräftige Unterstützung vieler Menschen nicht möglich gewesen. Mein Dank gilt zuallererst Prof. Peter Purgathofer, der mich betreut und seine Ideen und Erfahrungen eingebracht hat. Herzlich danken möchte ich auch meinen Interview-Partnern und den Kolleginnen und Kollegen an der TU Wien, die sich als Testpersonen zur Verfügung gestellt haben. Ihrer Bereitschaft verdanke ich viele interessante Einblicke, geniale Ideen und spannende Verbesserungsvorschläge. Auch in meinem privaten Umfeld gibt es Menschen, denen ich sehr dankbar bin: Besonders meinen Eltern, die mich nicht nur finanziell, sondern auch mit Rat und Tat unterstützt haben, und meiner Freundin Giselle, die mich mit bewundernswerter Ausdauer unterstützt und motiviert hat.

Abstract (English)

This diploma thesis investigates the role of information technology to support meetings of a group of people in a meeting room. Although research projects show that there is a need for improvement in the usage of technology in meetings, very few tools actually support presenting and sharing digital content, which is an important part of today's meetings.

Preed, the setup accompanying this work, consists of two parts: The software provides a web based user interface for the participants of a meeting. Using this interface, the participants are able to share digital artefacts like websites, images, notes and documents with each other. They are also able to present these artefacts on the presentation screen, without any action of the presenter. The hardware setup of *Preed* provides an HDMI-interface to the presenter, allowing him/her to connect their laptop and show a presentation or other digital content. The use of *Preed* does not require any installation on the hardware of the meeting participants.

During the design and development of *Preed*, ethnographic research and sketching played an important role. It was shown that thorough research and user involvement help in the understanding of the actual usage scenario as well as in making design decisions.

Abstract (Deutsch)

Diese Diplomarbeit untersucht, wie Informationstechnologie eingesetzt werden kann, um Besprechungen einer überschaubaren Gruppe von Personen im gleichen Raum zu unterstützen. Obwohl wissenschaftliche Untersuchungen zeigen, dass es gute Gründe für die Verwendung von Informationstechnologie in Besprechungen gibt, existieren im Moment nur wenige Systeme, die das Präsentieren und Verteilen von digitalen Inhalten unterstützen.

Preed, die Applikation, die im Rahmen dieser Diplomarbeit entstanden ist, besteht aus zwei Teilen: Der Software-Teil ist eine Web-Applikation, die den Teilnehmern einer Besprechung zur Verfügung steht. Mit dieser Web-Applikation können die Teilnehmer einander digitale Artefakte wie Websites, Bilder, Notizen und Dokumente weitergeben. Diese Artefakte können sie auch am Präsentationsbildschirm präsentieren – ohne Zutun des Präsentators. Die Hardware von *Preed* stellt dem Präsentator eine HDMI-Schnittstelle zur Verfügung, an die er sein Notebook anschließen kann, um eine Präsentation oder andere digitale Inhalte zu zeigen. Um *Preed* zu verwenden, muss keiner der Beteiligten Software auf dem eigenen Gerät installieren.

Beim Design und in der Entwicklung von Preed waren ethnographische Methoden und das Zeichnen von Skizzen von großer Bedeutung. Aufwändige Recherchen und die Einbindung von Benutzern halfen mit, den Kontext der Verwendung von *Preed* zu verstehen und Designentscheidungen zu treffen.

Table of Contents

1 Introduction

Preed, the application accompanying this work, allows the sharing, publishing and presenting of digital artefacts in meetings. The name Preed is derived from the words “presentation” and “feed”. This chapter will describe the motivation and goals for the development of this tool.

Starting in October 2008, I attended a university course named “Theory and Practice of Media Design“. This course consists of several day-long workshops and assignments that are to be completed in the between two lectures. At the beginning of each workshop, the assignments of the approximately 15 students are reviewed by the lecturer and the colleagues: Each student presents the result of the assignment, mostly a digital artefact like a digital image, a PDF document, slides of a presentation or a video.

A typical workshop does not only include the presentation of the assignments and the presentation of new slides by the lecturer, but also phases of research: The mission is to collect material concerning a special topic, e.g. the design of ATMs. The students then have a certain period of time (typically 20 minutes) to collect material like images, videos, documents and websites individually or as a group. This material is afterwards also presented to the whole group.

Of course, as the talk of the lecturer progresses, the students bring up digital artefacts in order to respond to questions or to illustrate points they want to make in the discussion.

During the course, it quickly became obvious that sharing und presenting this kind of material is a painful process. Either the lecturer or one of the students connects their laptop to the HDTV screen in the institute’s library where the course takes place. When a different participant wants to present content on the screen, there are (in a standard setup) different options. Each of them is unsatisfying in a different way.

The laptop of the student can be connected to the HDTV screen with an HDMI or DVI cable. If the considered laptop has such a connector, a sufficiently long cable has to be provided. The lecturer has to unplug the cable which possibly resets the carefully arranged windows on their extended desktop. The student has to plug in the cable and configure their computer according to the resolution of the screen. This procedure takes a minimum of 30 seconds– too long to be executed for a quick glance at a picture.

Alternatively, the content can be transferred to the presenter’s laptop. This might happen via a physical medium (USB flash drive, external hard drive, etc.) or via a network of some kind, using standard software installed on both machines (file sharing, IM software e-mail, etc.). The physical options require manipulation; for use of a network, software has to be installed and often the knowledge of the presenter’s account name is necessary. URLs will not be transferred as files, but will be spelled out or directed to via a search engine like Google.

These approaches are cumbersome, slow and time-consuming, especially compared to pushing a sheet of paper over the (real-world) desktop or draw a sketch on a whiteboard.

Another aspect should not be forgotten: It requires a considerable amount of work to collect and archive all of the artefacts that where visible to the participants during the course. A collection of the lecture material is possibly important for the next assignment or other purposes.

With these experiences as a starting point, a hard- and software setup was to be developed that allows sharing und presenting digital artefacts in the context of such a course. Only a few guidelines were set up in the beginning to lay ground for the design and development.

The setup should provide a shared, persistent workspace for the students and the lecturer in the same room. The workspace should focus on presenting and sharing artefacts – not on co-working on the same artefact. There are specialized tools and research projects for editing different media types collaboratively. These tools might be used in the context of the workspace, but the new setup should not aim at replacing them.

Further, the users should be able to continue working with the devices they are used to, like laptops, tablet computers, and smartphones. This is backed up by researches like Hiroshi Ishii, an MIT professor concerned with groupware systems, who stated in 1984 that "groupware that asks them [the users] to abandon their familiar tools, methods, and even computer hardware and software, and to learn a new system [...] is likely to encounter strong resistance." [1] Especially students of computer science, who can be expected to be highly skilled when working with a computer, might feel uncomfortable when forced to use a different system. Research also indicates that setups with a combination of networked computers and a shared screen show better results than others. [2]

In addition to that, the setup should work out of the box with a computer and standard software like a web browser. Software installation requires preparation and raises the barrier to use the system. In contexts other than university, it also requires the users to have the proper permissions granted by the system's administrator. For the ease of use of content sharing, the application should allow operations like drag & drop and copy & paste from standard software.

To the presenter, the setup should not only require no installation, but also be as transparent as possible. Both, the course lecturer and a student who happens to present an assignment, should be able to connect their computers to a provided interface without any hassle.

Not only installation but also configuration effort should be kept as small as possible. CSCW systems like BSCW [3] provide powerful options for granting permissions for every single user action. However, in the context of a small course with a manageable number of students in the room, restrictions can also be put up through social interaction instead of technical measures. It is more appropriate to tell a student not to publish certain kinds of content than to configure the system so that the student is not allowed to do so. The application itself should treat the presenter and the listeners equally.

For the optimal use of the commonly used screen in this setup, it should work with a screen at full HD (1080p) since the institute's library where a lot of workshops take place features such a device.

The aim of these guidelines is to keep the barrier to use the setup as low as possible. It does not have to be agreed on using it, it becomes "roomware" [4] like a table, an HDMI screen or a video projector.

2 Thesis structure

Following this introduction, examples of relevant related work in the field of computer supported collaborative work and commercial software products will be presented in chapter 3.

Chapter 4 will describe which authors and lines of thought have influenced the design and development process of *Preed*. Their key propositions will be presented. Additionally, their applicability to software development in a university context will be discussed.

Chapter 5 gives in an in-depth view of the different steps in the design and development process of *Preed*. Several ideas were explored in the course of the work; all of them will be presented alongside with numerous sketches and screenshots as well as reviewed retrospectively.

The anatomy of the final product will be described in chapter 6. First, a detailed description of the user interface is given, followed by an overview of the reference setup and implementation that were used to realize a prototype.

Chapter 7 outlines developments that might be done in the future based on the work at hand.

3 Related Work

The scientific field of Computer-Supported Cooperative Work (CSCW), a term coined by Iren Greif of MIT and Paul Cashman of Digital Equipment Corporation as a result of a workshop held in 1984, deals with information and communication technologies that support collaboration between individuals and groups [5]. Clearly, *Preed*, the system presented in this thesis, belongs to this multidisciplinary field.

A widespread classification system for CSCW systems is the time/space-matrix by Robert Johansen [6]. He classifies cooperation situations along two dimensions. The x-axis states whether the collaborators work synchronously (i.e. at the same time), or asynchronously (i.e. at different times). The y-axis takes into account where the collaboration takes place: It can happen in the same place, e.g. in the same meeting room, or in different places. With this definition, a two-by-two-matrix is built:

		Time	
		synchronous	**asynchronous**
Place	**co-located**	Electronic whiteboards, shared tables, wall displays, roomware, …	Meeting rooms, large public displays, …
	remote	Video conferencing, instant messaging, chats, screen sharing, shared whiteboards, multi-user editors, …	E-mail, group calendars, workflow tools, version control, …

Table 1: Examples of CSCW systems [7]

The following sections describe related work in the top left hand corner, i.e. CSCW projects that focus on supporting cooperative work which takes place in the same place at the same time. Both selected research projects and commercial software products are presented with respect to the needs that were presented in the previous sections.

3.1 Project Nick

Project Nick put research effort into meeting analysis and augmentation as well as technological support before, during and after a meeting. The researchers at the "Microelectronics and Computer Technology Corporation" (MCC) in Austen, Texas, were exploring "the early part of the design process, before requirements are established, for large-scale distributed systems" [8]. Since meetings were necessary elements of their projects, the researchers were interested in analyzing their structure and dynamics, finding a proper vocabulary for it und in the end develop technology to support meetings.

It was noticed that guidebooks for conducting meetings introduced different methods, but did not advise the use of technology. However, the researchers believed that "the introduction of appropriate technology is an important component of our work" and that "technology can positively change meeting culture if it is introduced and used in sensitive, socially responsible ways" [8].

Anticipating Johansen's classification of CSCW systems [6] they described that meetings can be classified by means of time and space and that each of the four combinations has specific needs and requirements.

By observing meetings of MCC's design teams, *Project Nick* developed a theory of meetings: It consists of an octet, M = {G, R, P, I, A, S, C, E}, where the components are the following:

- "G: the set of **g**oals of the meeting,
- R: the set of **r**esources used within the meeting,
- P: the set of **p**articipants of the meeting,
- I: the **i**nformation manipulated within the meeting,
- A: the set of **a**ctivities that transpire during the meeting
- S: the meeting **s**tructure including meeting type, agenda, and rules of order
- C: the **c**ontext in which the meeting takes place, and
- E: **e**xternal factors that only bear peripherally upon the meeting." [8]

Technically speaking, the actions in the context of a meeting can be thought of as a function with I, P and R as inputs. The function executes a conversion described by A, which is influenced by C, G and S. All information of the meeting resides in a repository called "noumenon"[1] (N). No single participant has access to all of this information, but for every participant there exists a mapping for N, namely the personal knowledge of this person. The better the communication in the meeting, the closer are the N-mappings of two participants.

After establishing this theory of meetings, a number of electronic equipment was developed and explored. In the meeting room of *Project Nick*, every participant has a private workstation, connected via LAN to the other workstations and information servers outside the meeting room. The workstations are equipped with group work software to display the agenda and provide access to shared content. An electronic whiteboard is also present in the room. This can be used for sketches or presenting content of the participants' private screens.

[1] A „noumenon" refers to information that is known, but not subject to perception; in contrast to a phenomenon which can be discovered by the human senses.

The researchers classified content along two dimensions: To whom the content is visible and how it is structured:

	private	**subgroup**	**public**
binary	list check off	support flags	voting
structured	lists	facilitation messages	presentations, lists
unstructured	bit pad	notes and comments	electronic board sketch

Table 2: Meeting information matrix [8]

In several experiments, the exchange of these types of information was explored. For example, the facilitator of the meeting could receive mood messages like "I am bored" or "the meeting is very interesting" from the participants. These messages should help the facilitator to propel the meeting. The software also provided for taking notes, voting, sketching, subgroup communication and content sharing.

3.2 CoLab

In the late 1980s, researchers at Xerox PARC's Knowledge Systems Area were frustrated because after lively brainstorming sessions on whiteboards they had to capture all of the information again on their computers. As a result of this frustration, the *CoLab* Project was founded [9]. In this setup, every user has their private workstation to access personal data and documents relevant for the current meeting. On a rear-projected digital whiteboard visible to all of the users, content can be shared and edited. A user who is working with their personal workstation has the option to make a window publicly visible on the whiteboard. This concept is called "public windows" by the authors. In the beginning, the content on the public display could only be manipulated using a separate mouse and keyboard. With the development of the *LiveBoard*, a large interactive screen which is described in the next section, interactions were possible with a cordless pen [10]. A *CoLab* did not only support meeting participants with technical equipment, but also provided the researchers with a separate workstation to record and analyze the meeting events.

However, the researchers noticed that participants of spontaneously initiated meetings did not go off to the *CoLab* which seemed to be too much effort [11]. They developed ideas to integrate *LiveBoards* in everyone's office in order to conduct "portable meetings".

One tool that was developed as a part of the *CoLab* Project was *Cognoter*, a tool to generate and cluster ideas. From today's point of view, it was a predecessor of brainstorming software: Nodes could be added, moved, commented, connected and deleted. Many *CoLab* applications supported a pattern of three phases named "brainstorming – ordering – evaluation" [12].

In total, four such *CoLabs* were built and numerous applications were developed. PARC itself considers the *CoLab* project as a milestone of the company's history [13].

Figure 1: First CoLab (left) and one of its successors in Webster, New York (right).

3.3 LiveBoard

An interesting component of the *CoLab* is the *LiveBoard*. The researchers in the early 1990s noticed that a large and central display is an important factor in electronic meeting support systems [14]. At that time, they were mainly used for presenting content. Projects like *VideoWhiteboard* [15] supported remote collaboration on a whiteboard, but dedicated systems to support face-to-face meetings were not available.

The aim of the project was to develop a "directly interactive, stylus-based, large-area display for use in computer-supported meetings" [14]. Since large flat-panel screens were not available, the scientists decided to use a liquid crystal display (LCD) and rear-project it to a sufficiently big screen. This has two main advantages; first, the projection does not show any jittering, a problem that often occurred when projection systems were based on a cathode ray tube (CRT) based screen. Second, in contrast to an LCD, the rear-projection is independent from the angle of view. This is very important since unlike in a cinema setting, the participants of a meeting need to see the content on the screen from everywhere in the room.

For interaction with the content on the screen, a cordless pen is used. The researchers anticipated that several users might interact with the *LiveBoard* at the same time, which would have raised problems with a wired solution. The pen can not only be used in direct contact with the *LiveBoard*, but also for pointing to content or the use of gestures. Inspired by a three-button-mouse, the pen also had three buttons. The pen emits a ray of light which is analyzed by a detection module behind the rear-projected screen. In this way it can be used from a distance.

The *LiveBoard* provides interfaces ("planks") for different application scenarios; in addition to the default meeting application (featuring a whiteboard, a text editor and a clock) there are interfaces for games and slideshows. For the latter, the *LiveBoard* can display a multi-page postscript file as a slideshow which is controlled by gestures of the presenter with the cordless pen. Scaling and scrolling are also controlled by means of the pen, writing on the slides is possible, too.

Evaluations and observations showed that the resolution (1120x780 pixels, projected to 116x81 cm) on the *LiveBoard* and the accuracy of the pen were insufficient. In addition to that, the use of the three buttons on the pen was avoided by the users. Interestingly, a group of users experiencing a technical problem was less willing to solve it than they might have been as individuals in front of their workstations. They would rather use a conventional whiteboard.

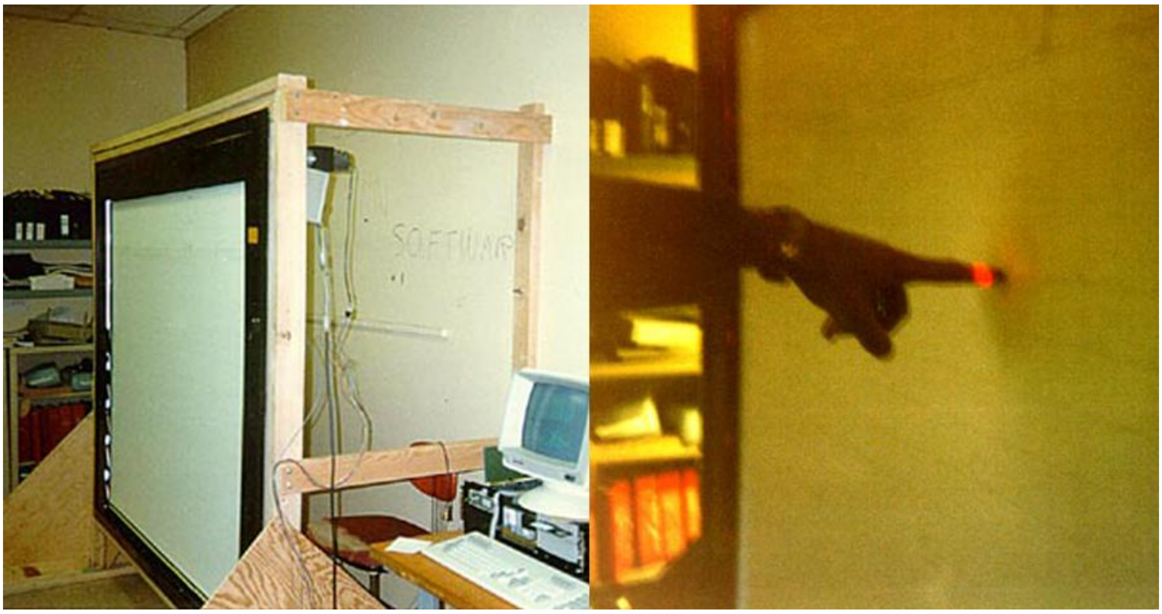

Figure 2: A LiveBoard (left) and a human interacting with it (right).

3.4 DOLPHIN

A similar setup is used by *DOLPHIN* [10], a project by the German National Research Center for Computer Science, which is part of the Fraunhofer Society today. Like *CoLab*, *DOLPHIN* aims at providing tools for a broad range of activities and does not focus on one specialized task like collaborative writing or brainstorming. The researchers observed staff meetings of the editorial board for an electronic newspaper to investigate the requirements.

However, the face-to-face-setting was extended to four supported application scenarios:

- The public screen named *LiveBoard* can be used by a single user with a pen.
- Several users might use the *LiveBoard* and the workstations in the room. This scenario applies to the meeting of the editorial board.
- In addition to that, individual users might be connected with their workstation. This adds complexity since they have to have access to a synchronous communication channel like audio and video. Also, these users need to have access to the content and the interaction on the *LiveBoard*.
- Alternatively, a second, similarly equipped room might be linked. Like in the previous scenario, a synchronous channel has to be provided. The *LiveBoards* in both rooms share the same content in this application scenario.

DOLPHIN's document model uses nodes to abstract different kinds of content. Texts, scribbles and images can be linked and grouped; a node might contain all of these entire object types at the same time. This structure is flexible enough to support formal and informal structures. Scribbles and notes can be taken on the *LiveBoard* (informally) before they are brought into a formal structure later on in the meeting. In this way *DOLPHIN* does not – contrary to *CoLab* – force the users into a pattern the meeting has to follow.

From the observations of meetings, it became clear that participants have the need for private notes. *DOLPHIN* therefore supports public and private workspaces: On each workstation, a document model compatible to the one on the *LiveBoard* is displayed. Users can take notes or prepare content on their private workstations and then copy the content to the *LiveBoard* and make it visible to every meeting participant. Since every participant is provided with a private workstation, actions that derive from the meeting and only involve one participant – like writing an e-mail to another colleague – can be carried out immediately.

The *LiveBoard* plays an important role in this setup: It is the main focus of the participants and offers functionality to manipulate the shared content. Gestures carried out with the cordless pen control actions on the *LiveBoard* like creating, moving and deleting nodes. Drag & drop operations are supported for the grouping of nodes.

The system's architecture is a client-server-model, where the *LiveBoard* as well as the workstations are connected to a media server via LAN or WAN.

3.5 i-LAND

Starting with a sketch (Figure 3) in 1997, *i-Land* is an "interactive landscape for creativity and innovation" [4]. It is part of the "roomware"-concept by Norbert A. Streitz and his colleagues at Fraunhofer's Integrated Publication and Information Systems Institute (IPSI). Streitz argues that for further advancement, the field of Human Computer Interaction should turn its attention to architecture. Interaction between humans and computers is not a means to an end, but only an intermediate step between the human and the information he wants to access. Therefore, the focus should lie on human-human and human-information interaction. The researchers believe that computers will be invisible in the future. Computer augmented objects and cooperative buildings will be the interface to the information humans want to access. In such cooperative buildings, interaction with information will not be restricted to meeting rooms.

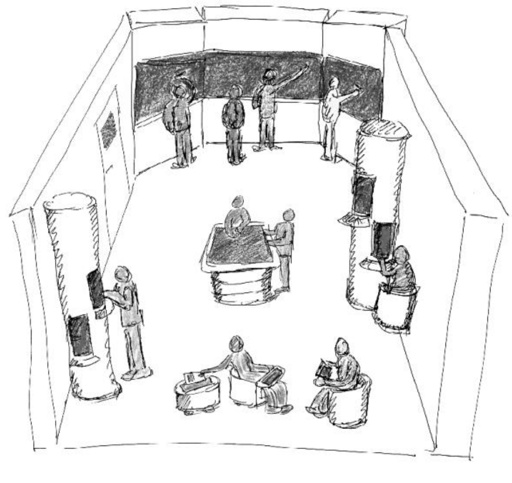

Figure 3: Sketch of i-Land [4]

In this context, *i-Land* is the environment in which these ideas are envisioned. Chairs, tables and walls of this environment are built to support rich interaction with content in order to create innovative ideas. The environment consists of different components which will be presented in the next paragraphs.

DynaWall is the large version of the electronic whiteboards mentioned in the previous projects. The interaction is based on gestures; several users can interact with the screen at once. Because the industry does not offer very large affordable displays, three screens are mounted together to a total width of 4.5m. This allows for new interaction paradigms: Dragging an object from one end of the display and dropping it at the other is not desirable. “Take and put” as well as “throwing” objects around the wall are ways to cope with this problem.

In *i-Land*, the meeting participants take seat in armchairs called *CommChairs*. These computer augmented pieces of furniture provide computers (or docking stations) for their users, who are then able to take private notes and to interact with public workspaces. Another moveable component is the *ConnecTable*, a kind of cocktail table with a mounted display. These tables can be put together to form a bigger display on which two or more users can interact with content.

Standing around an *InteracTable*, a horizontally mounted display without a predefined orientation, the users can display and discuss content by drawing and annotating it with a pen or interacting with their fingers.

Figure 4: A marketing shot of i-Land [4]

The researchers use an interesting approach for transferring information objects from one component to another. Since the components are equipped with scales, a broad range of objects like a bunch of keys or a watch can be used. It has to be put on the designated area on the component. An information object can then be dropped on the object. The system only remembers the weight of the information carrying object. When it is dropped onto the scales of another component, the information object becomes available there. Alternatively to weight, RFID tags can be used as an identifier for the information carrying object.

3.6 Netop Vision 6

This classroom management software was formerly named *MasterEye*, which describes where the focus for this kind of software lies: On control for teachers. *Netop* [16] advertises that the software helps to keep the students focused on the content of the course by eliminating distractions by prohibiting them: The students' screens can be blackened out by the teacher, access to applications and websites can be restricted. The teacher can supervise the activities of the students on their computers while they are in class. *Netop* Vision 6 provides tools for showing the teacher's screen on the screens of the students and vice versa. Collaborative web browsing and options for collecting completed tasks are also included. Live collaboration is fostered by remote assistance tools and a messaging system. Also, commands can be sent to the students' computers, e.g. for opening a document from a remote location in a word processing software.

The application has to be installed on every computer participating in the network of the class. This is possible in a setting of wired computers in a lab or laptops that are given to the students by the school. However, it is not suitable for a university or business setting, where participants cannot be forced to give up control over their computers. Even if they were ready to this for the sake of some new features, installation and configuration would form a high barrier to using such a system.

3.7 Netviewer

Remote desktop sharing software like *Netviewer* [17] are developed for settings where the participants of a meeting are not in the same location. Consequently, this software has a lot of built-in features to cope with dislocated participants like VoIP, videoconferencing, text chat and file transfer.

In order to participate in a meeting, the user enters a session number and their name on *Netviewer's* homepage. For the fully featured software, download and execution of an executable file is required. A flash-based client with a smaller feature set and lower performance is also available. The moderator needs to have a different software package installed, where he is able to start and moderate a session.

Similar to classroom management software, the moderator and the participants are able to broadcast their screens and take over control over a certain computer. In contrast to classroom management software, these actions have to be confirmed by the affected user. *Netviewer* offers detailed options of controlling the visibility of different applications. Both the moderator and the participants can share files and – assumed suitable software is present on the user's computer – open them. However, the user is responsible for archiving the files transferred during the session since they are no longer accessible once the client software is closed.

For presentation purposes, users can scribble on the screen of another user or point to important elements with a large coloured pointer.

For co-located live collaboration, *Netviewer* is an option. However, it lacks support for a large public screen. This could be solved by connecting the presenter's laptop to the public screen and mirroring their screen on the public screen with the help of the operating system. However, it is unlikely for a student to have the software installed. In addition to that, handling the software is not always straightforward due to its large feature set.

3.8 SMART Board

Digital whiteboards like the *SMART Boards* by Smarttech [18] are promoted with their collaboration features. The *SMART Board* is a combination of a short-throw projector and a canvas that is installed in a classroom. The projector is mounted above the canvas and is equipped with a special lens to avoid distortion. A dedicated component, housed together with the projector, tracks the interactions on the board.

The students can interact with the board in two ways: Pens in different colours are provided; these can be used to draw and write on the board. Alternatively, the board allows for multi-touch: This way, one or several students can use the board at the same time.

A variety of content can be shown on the *SMART Board*: Images, drawings, videos, 3D content, presentations and interactive applications are possible. In the case of the latter, students can, for example, drag and drop the planets of our solar system in the right order or build a simple machine out of cog wheels. To share content on the board, a separate application is available for presenters.

However, it is not possible for other users than the presenter to share digital content on the board. Additionally, the paradigm of direct manipulation is difficult for groups as soon as they are too big to gather around the board.

3.9 JamJar

JamJar [19] is a demonstration application by Adobe Labs. The web application is built on the Flex framework. After logging into a workspace, the concurrent users see an infinite two-dimensional canvas. On this canvas, the users can create and import different types of content like texts, links or images. The canvas is continuously updated so that every user gets notified of recent changes.

The infinite canvas paradigm is an interesting approach for visual content in a creative environment: Content can be imported, arranged, grouped, moved and annotated. JamJar takes the idea of digital whiteboards one step ahead by distributing the infinite canvas to all of the clients' computers. The users can then edit the content individually with the power and the tools of their computers and upload it again.

However, for the intended use, *JamJar* has several drawbacks: Apart from not being available to new users anymore, *JamJar* does not provide for drag & drop operations from outside the browser window. Furthermore, JamJar only supports limited file types: It is not possible to upload PDF or Word documents. Finally, the bigger the canvas gets, the more challenges arise with retrieving a document again. Like all of the other software products, JamJar does not provide any features for a large public screen.

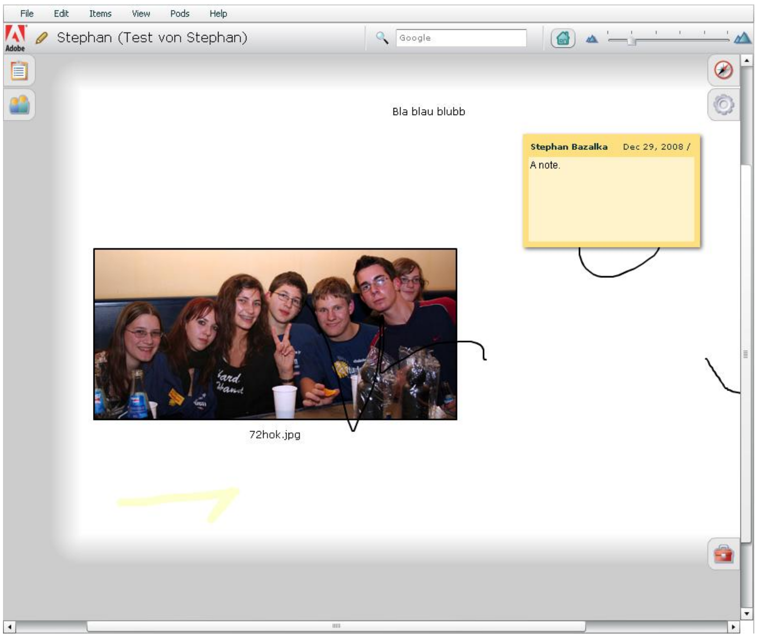

Figure 5: User interface of JamJar

3.10 Conclusion

Interestingly, researchers discovered very soon that the use of computers offers new possibilities for meetings. Projects like CoLab were the first to make suggestions in this field. In the meantime, technology has changed, but interaction has not. Still, participants of meetings are struggling to exchange data. Solutions like *Netop Vision* live in a very narrow niche; products like *Netviewer* are made for different purposes. Especially the latter could be used for exchanging files. But it lacks features for collaborative screen control and for logging of the artefacts.

As for the problem described earlier, a new solution has to be found.

4 Method

This chapter gives an overview of the methods of software product development that have influenced the development of *Preed*, the application accompanying this work. Numerous schools of software development exist, but the ideas of two forerunners will be presented. On the one hand, Alan Cooper, the "Father of Visual Basic", developed a method called "Goal-directed design". His personas concept, very precisely described, prototypic users, has been widely adopted. On the other hand, Bill Buxton, a former Xerox PARC researcher, described how a flexible design process and the power of sketching can help craft good user experiences. After a short overview of the history of software development for graphical user interfaces, the thoughts of both authors will be presented. First, with the considerations of the software development process as a whole, followed by suggestions for the design steps in such a process.

Software engineering for graphical user disciplines is a comparably young subset of engineering, given the achievements of civilizations like Ancient Egypt. The first graphical user interfaces were developed in the 1970ies but did not have their breakthrough outside of a computer-savvy community until the early 1990ies.

Several methodologies have been proposed for software development. A widely known and adopted model is the *waterfall model*. In 1970, Winston Royce presented a process consisting of discrete steps: System requirements, software requirements, analysis, program design, coding, testing and operations. Each step has a feedback loop to the previous and the consecutive step. Under the premise "Do it twice", he also introduced a concept which is today referred to as prototyping: "If the computer program in question is being developed for the first time, arrange matters so that the version finally delivered to the customer for operational deployment is actually the second version in so far as critical design/operations areas are concerned. (…) Note that it is simply the entire process done in miniature, to a time scale that is relatively small with respect to the overall effort." [20] – However, the waterfall model has severe disadvantages. Especially when it is implemented without the feedback loops, big challenges arise when not every single requirement was documented in the corresponding step. Some experts even argue that it is simply built upon false assumptions like that supporting a belated software project with additional developers will speed up the development [21].

Several adoptions to the waterfall model have been proposed as well as different process models, namely iterative models like the spiral model by Barry Boehm [22]. Still, these models are geared to a traditional point of view of engineering (a software factory) and did not satisfy the 17 authors of the *Agile Manifesto* [23]. Principles like "Working software over comprehensive documentation" and "Responding to change over following a plan" are a definite rejection of the processes in waterfall-like models. The signees of the manifesto also value "Individuals and interactions over processes and tools" and "Customer collaboration over contract negotiation". To support these ideas, several methods were developed. One of these is the concept of user stories, short examples how the desired product will be used, which aim at bringing a more user-centric approach the development.

However, several authors have recognized that the process model is not the only determining factor whether the resulting product is perceived useful and desirable by the prospective users. Two of these authors are Alan Cooper and Bill Buxton.

In his book *The Inmates Are Running the Asylum* [24], Alan Cooper lays the foundations for *Goal-directed Design*, a method for interaction design. Cooper identifies several reasons why computers and information technology is difficult to use:

- The goals of the developers and the goals of the users differ heavily. The first aim at ease-of-programming, the latter at ease-of-use, but the programmers hold the whip hand. Since their activity is very challenging and difficult, the requirements of the users fade in contrast to the requirements of programmers [24]
- Since developers are not given clear guidelines, they stick to the principles and elements which they know, the ones they like the most or the ones that are coded easily. [24]
- Even if they are given clear guidelines, developers seem to take these as suggestions. To outline how particular this behaviour is, Cooper gives an example from architecture. A future house owner expects the construction workers to follow the architect's plans to the word (or to the ruler) and will not be willing to accept a row of two windows instead of three because the workers were able to reuse prior work. [24]
- Typically, the developers are not the users of the prospective system, so their view on the product is biased heavily. Typical developers are ready to accept complexity as a trade-off for more control, whereas a typical user wants simplicity and accepts less control as a trade-off. [24]
- When it is accepted that design is of high importance for the product, developers expect it to be done after they are finished. [24] Cooper calls this "painting the corpse". [24]
- Managers have made the experience that while using the same process, some products are a success while others turn out to be a failure. They have concluded that this is only a matter of luck, not a matter of the application of interaction design. [24]

It is laid out clearly that the programmers are not to be blamed for these problems. They have been trained and educated in various skills, but not design. Cooper concludes that "we have to revamp our development methodology so that the humans who ultimately use them are the primary focus." [24] When compared to the construction to a building, interaction designers act as architects, not as interior designers. [24] Cooper argues that a process step dedicated to design prior to programming will render software products more usable and desirable.

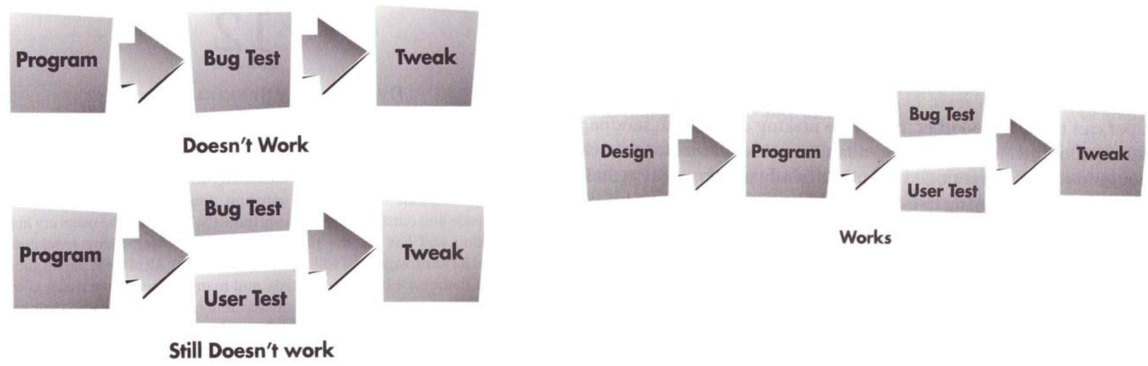

Figure 6: Not successful software processes (left), suggested process (right) [24]

As shown in Figure 6, Cooper still relies on user tests, but only as a supporting step after programming has taken place.

For the design phase, Cooper suggests a method called *Goal-Directed Design*. It consists of several artefacts related to and derived from each other, which finally lead to a detailed specification of the final product. So detailed that designers and not programmers have the control of the result.

As a start, personas are defined. Personas are prototypical users, derived from knowledge about real users and their problems. This knowledge can be gained with ethnographic methods like interviews and observation. The resulting fictional users will be used to have a specific image of the how the product will be used and what the user will love about it. In conversations, sentences like “the user will do this” are often used to justify personal preferences, whereas a well described persona will help answering design decisions. To help the designers work with the personas, they are given a name, a face, an occupation, an age and other very specific characteristics so that the persona becomes very concrete for the designer. While similar considerations are made in market research, a persona always refers to a person who actually uses the product – not the person who buys it.

Apart from their personal characteristics, personas are defined by their goals. A product is not used as an end in itself, but because it serves a persona’s goal. It is important to note that goals are different from tasks. Several tasks are carried out to reach a goal. This in mind, a goal is to read a book, and the corresponding tasks are to select a book and buy it. While tasks change as technology changes, goals do not change accordingly.

Ethnographic methods like interviews, observations and field trips have been explored by a number of researchers. These methods can be used to gain knowledge about the personas and their goals. Jeanette Blomberg and others in [25] have described the guiding principles for ethnographic work:

- Natural Settings: Ethnographic work is interested in observing people in the context of their daily lives or daily work. In contrast to experimental studies or observations in the laboratory, this kind of research takes place in a real-life setting.
- Holism: It is believed that certain principles and mechanisms can only be observed "in the wild", i.e. a natural setting, while they stay obscure in a laboratory.
- Descriptive: The result of ethnographic work is a description of human behaviour. The observed actions are not judged or evaluated.
- Member's Point-of-View: Ethnographic research wants to understand the world from the point of view of the observed people.

Ethnographic research is interested in gaining an understanding of how people work, live and percept the world around them. Since users are typically not involved in technology development, ethnographic work is an important part of understanding the prospective users problems and allows designing for a solution of this problem.

Coming back to *Goal-Directed Design*, interaction design still has to deal with tasks. In *Goal-Directed Design*, this is done with the help of scenarios. These are precise sets of actions that need to be taken in order to achieve a goal. Commonly, these tasks are already recognized during interviews or observations of prospective users, but not put together until the set of personas is complete. Scenarios are closely related to personas, since scenarios describe how a certain persona interacts with the desired product and the rest of his/her environment. In general, scenarios are crafted without respect to technical limitations or constraints. Daily-use scenarios are most important and need thorough consideration since they will be experienced very often by the users. Necessary-use scenarios define actions that must be possible, but are not carried out frequently. Finally, edge-case scenarios complete the set of scenarios with descriptions of situations that occur seldom but need to be considered in order the cover all cases.

From these scenarios, a detailed specification for the product can be derived and handed over to the programmers.

Bill Buxton is also a strong supporter of a stronger role of design in software product development. As he describes in his book "Sketching User Experiences" [26], the software product development often seems to consist of only two consecutive phases: Engineering and sales. He considers this approach to be "broken" [26]. The conceptual models and interaction principles related to computers have been the same for decades. Even worse, they are being repeated regarding the development of emerging technology like smartphones. He argues to put more efforts "to understand the larger social and physical context within which it [the product] is intended to function" [26].

Buxton also shows that software companies are bad at developing new products. One of the world's largest players, Adobe, has only developed a single software product (Acrobat) after entering the market with Illustrator. All other products are the result of mergers and acquisitions. Maintaining an existing software product is also very hard:

- Complexity grows with every release and when maturity is reached, it is very difficult and costly to rewrite or refactor the product's code.
- If the product is successful, the number of installations will reach the critical mass at some point. Meaning that innovations do not only have to please new customers, but also the existing ones.
- The longer the product exists, the more have the obvious features have been added and customers who are won most easily have already been convinced.

Concluding, while the costs for maintaining a software product keep growing, it is getting more and more difficult to gain more market share and keep users updating to newest version. This being the case, Buxton points out that the process to develop new products and services is a crucial factor for a successful company.

Still, design is obviously not devoted enough time and resources to. Like Cooper, Buxton compares software product development to film making: It is the aim of preproduction to get a clear, detailed view of the final product and to plan production as well as postproduction. While there is still room for creativity during production, most of the design decisions have already been made during preproduction. Furthermore, the use of preproduction is to reduce production costs. Consequently, Buxton asks "how is it that we can never afford to do proper planning and design, yet we always seem to be able to afford to pay the cost of products being late as well as the cost of fixing all of the bugs that inevitably result from inadequate design, planning, and testing." [26]

Apart from planning, another aspect has to be highlighted. As Donald Schön pointed out in 1983 [27], two different aspects of design have to be distinguished: Problem setting and problem solving. Problem solving refers to the "how" a problem is solved, i.e. how the development process is designed, how programmers work, how the product is marketed. Contrary, problem setting tries to answer the question "what" kind of problem actually exists and what a solution looks like – disregarding the actual process how this will be achieved. To sum up, a perfectly built product will still fail if it is in fact the wrong product.

Contrasting to Cooper, Buxton is not suggesting a self-contained design phase prior to engineering. Buxton claims that there are three phases but that they cannot be separated, meaning that the skills of design, sales, engineering as well as management and marketing are of different importance in the different phases, as shown in Figure 7. While all roles are involved in every phase, one specific role has the lead in every phase.

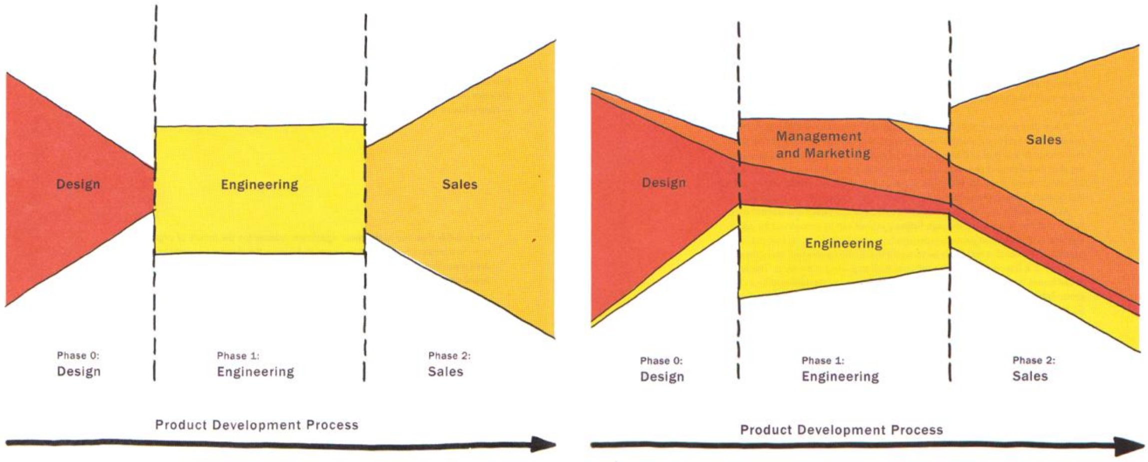

Figure 7: Three separated phases (left) versus an integrated view (right) [26]

Although simply adding a phase prior to engineering is alluring, it is based on the assumption that the stakeholders are aware of the desired result at the beginning of the development process and that their knowledge is sufficient for building it. Buxton considers this path risky and claims that design is reduced to styling and usability in such projects.

During this design phase, Buxton considers sketching the most powerful and useful technique. The technique of sketching as a tool for working out ideas and communicating concepts was first documented in the medieval ages and has not changed until today. By sketching, a designer can quickly and cost-effectively explore, store and compare several ideas. In addition to that, sketches clearly show that they are unfinished yet and that feedback is still possible and welcome. Buxton lists the most important attributes of sketches; some of them are rephrased in the following paragraphs:

- Quick: It does not take long to make a sketch.
- Timely: A sketch can be produced the moment it is needed.
- Inexpensive: The necessary resources are negligible.
- Disposable: Since the invested effort is rather low, a sketch can be succeeded by another one or considered useless completely.
- Plentiful: Because of their nature, it is easy to make a lot of sketches.
- Minimal detail: Sketches only provide an overview, a rough idea of the product, everything else is distracting.
- Suggestive nature: Sketches rather open discussions instead of closing them.
- Ambiguity: On purpose, sketches are ambiguous and therefore different interpretations are possible, even for the drawer. [26]

Buxton considers sketching more a process than an artefact, meaning that the process of sketching helps explore and discover ideas but that the sketches have no value in themselves. Sketching helps to bring ideas out of the designer's mind onto paper and generates new knowledge by seeing and interpreting them. An important element of the use of sketches is their use as "social things" [26]. While a sketch already helps a single designer explore his idea, even bigger value is generated when the sketch is shown to and discussed with the team.

Authors like Carolyn Snyder [28] support this case. While she uses “paper prototype” to describe the artefacts that Buxton calls “sketches”, she too argues that paper and pencil are tools which help generate new ideas. Addressing designers, she says that “you don’t want to choose a tool that introduces additional and unnecessary ones [constraints, n. b.], especially early in the process when you’re trying to be creative.” [28] A sketched prototype not only helps to avoid nitpicky feedback because of its unfinished look, it also helps the designer to withstand the temptation to make a draft perfect and pixel-accurate.

Interestingly, architects also make use of a sketchy look. Architects use CAD software to draw a draft of a building, but use different software to make it look like a sketch to encourage discussion.

Buxton also differentiates sketches from prototypes. Although they have their role in a design process, Buxton states that prototypes tend to answer questions instead of raising new ones. By contrasting the attributes of sketches with the ones of prototypes, Buxton argues why prototypes are a useful tool in an early design phase which is concerned with ideation[2].

Sketch	→	Prototype
Evocative	→	Didactic
Suggest	→	Describe
Explore	→	Refine
Question	→	Answer
Propose	→	Test
Provoke	→	Resolve
Tentative	→	Specific
Noncommittal	→	Depiction

Table 1: Comparison of the attributes of sketches and prototypes [26]

Still, a designer needs to deliver a concept or a specification of some kind in the end. While elaboration with the help of sketching results in more and more ideas, reduction and decision-making is also necessary. These two contradicting necessities have to be both respected. Buxton refines a thought of Stuart Pugh [29], who solved this challenge by alternatively generating and discarding ideas (Figure 8). Pugh called these two activities concept generation and concept convergence. Interestingly, Alan Cooper identified two similar roles for the interaction designers in his consulting company, named “generation” and “synthesis” [30].

[2] Ideation is a term originating from a combination of “idea” and “generation”.

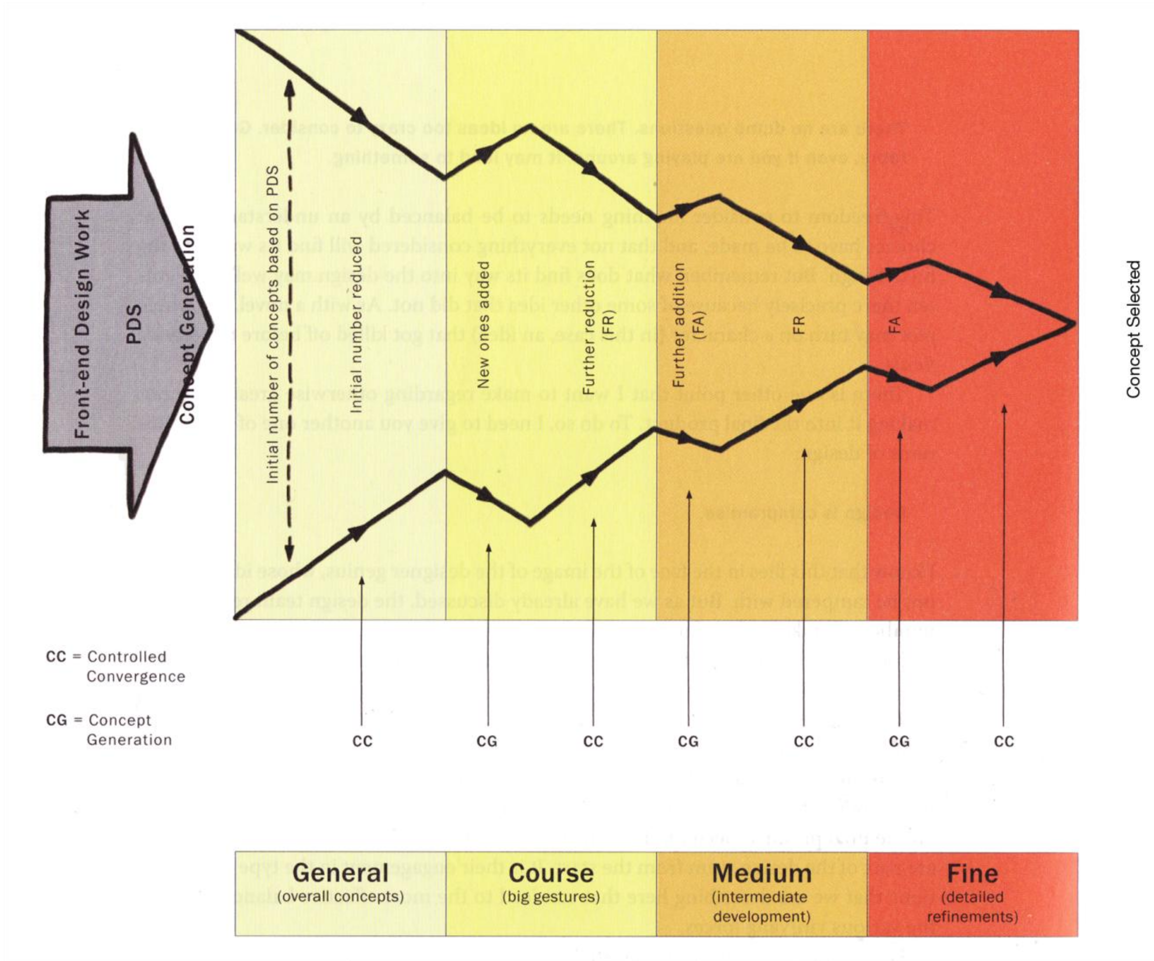

Figure 8: Alternating concept generation and controlled convergence [26]

An important point during this process is documenting the reasons behind design decisions, a design rationale. Knowing why certain decisions were made is helpful when a presumption changes along the path.

So far, the role of user in this process has not been investigated. Buxton argues that he takes the user's involvement for granted: "Arguing for the need for user involvement in a modern book on product design is as pointless as a discussion about the need to know the rules of arithmetic in an advanced mathematics textbook." [26] He does not suggest a certain method of user involvement, but wants to present techniques that are compatible to traditions like user-centered design or participatory design. Of course, techniques like sketching and prototyping allow user involvement at an early stage of the project.

In the end, one interesting agreement has to be pointed out. Both Cooper and Buxton agree that it should not be relied on a way breaking yet-to-come technology when designing a product. Buxton reveals that technologies like text editing or gesture recognition took decades of research until they were available in commercial product. He concludes that "we should not count on any deus ex machina. (…) It is highly unlikely that there will be any technology that we don't know about today that will have a major impact on things over the next 10 to 20 years." [26] Alan Cooper even puts it more negatively: "Microsoft says that interfaces will be easy to use as soon as it can perfect voice recognition and handwriting recognition. I think this is silly. Each new technology merely makes it possible to frustrate users with faster and more-powerful systems." [24].

CONCLUSION

In this section, an overview of two books was given. Both Alan Cooper's "*The Inmates Are Running the Asylum*" and Bill Buxton's "*Sketching User Experiences*" have influenced the development process of Preed heavily. It was shown how Alan Cooper considers the hegemony of developers over the development process as the basic problem why IT products are hard to use. His method, *Goal-Directed Design*, was presented. When following his propositions, personas being prototypical users ("personas") are derived from knowledge gained from ethnographic research. A persona consists of fictional biographic data and the goals this persona has with respect to the desired product. Detailed scenarios describe how the persona will use the product. Using this method, desirable products which put the user's requirements first, can be designed. Also, Bill Buxton's views of the product development process were presented. He also supports a strong role of design in the development process. His view of the process does not include self contained steps, but steps where one role dominates. With the help of sketching, ideation, concept generation and feedback are fostered. This happens to analyze the problem setting ("what is the problem?") in depth instead of diving into problem solving immediately ("how is the problem solved?"). Finally, Cooper's and Buxton's thoughts on the use of emerging technologies were described.

5 Design and development

This section will describe the process how *Preed*, the application accompanying this thesis, was developed. “Development” does explicitly not refer to the process of programming only but refers to the whole process, beginning from ideation over sketching, prototyping and testing to the final product.

As laid out in section 4, the thoughts of Alan Cooper and Bill Buxton have influenced the development of *Preed* heavily. Of course, huge differences exist between the product development process of a big software company and the development of a student’s project. The number of actors in the product development is very small and it is also clear that from the beginning on, no sales phase will take place. Nonetheless, the idea that the design phase is never finished and that sketching is superior to coding during the design phase is applicable to a student’s project as well. For this reason, the following chapter is more an overview of the different stages of *Preed* than a description of the final result or implementation details. Since the sketches are an integral part of the development process of *Preed* and its predecessors, several of them will be presented and commented.

For the production of the sketches, different tools were used: A simple *Moleskine* notebook and a pen or pencil was used for sketches showing an idea or a basic interface. More complicated user interfaces were done in *BalsamiQ Mock-ups*, an application for developing user interfaces that look hand-drawn. This hand-drawn look deliberates even perfectionist users from the necessity of working pixel accurate like in more powerful graphic software like *Adobe Photoshop*. When sketching interaction was more important than the actual interface, *Axure RP Pro* was used. This software allows the quick creation of clickable prototypes. Since there is no programming involved in the creation and it was only used to explore different interaction ideas and data views, these clickable HTML pages can also be considered sketches in the next sections.

5.1 First sketch

As described in the introduction, the impetus for this software application came from a workshop with a dozen of students who were challenged by sharing digital artefacts. Since the development of creative solutions was encouraged during the workshop, I started sketching a small but powerful application to make sharing easier (Figure 9).

In this first sketch, the lecturer creates a new workspace with a two-dimensional room plan based on the real setup in the room at the beginning of a workshop. The students then log in and occupy one of the virtual seats. Three different modes are supported:

- User to user: To exchange a file with one’s neighbour, a user drags a file from their computer into the application and drops it on the corresponding seat of the target user. It is instantly transmitted via some kind of wireless network and shows up on the screen of the target user.

- User to group: A certain area of the room plan is reserved for exchanging artefacts with the whole group. Artefacts which are dropped on this area become visible to all of the users in the current workspace.
- User to presentation: The presentation screen also has a drop-sensitive area in the room plan. An artefact dropped on this area will be instantly shown on the presentation screen.

This sketch focuses on sharing files. Consequently, their portrayal is similar to the one in operating systems with a graphical user interface. In the sketch, previews of the files are stacked on each other and can be accessed by clicking on this corresponding representation.

From a technical perspective, the users share and present artefacts with the help of a web application. Only the computer that is connected to the presentation screen runs a dedicated application responsible for downloading and displaying the artefacts that are dropped on the corresponding area in the web application.

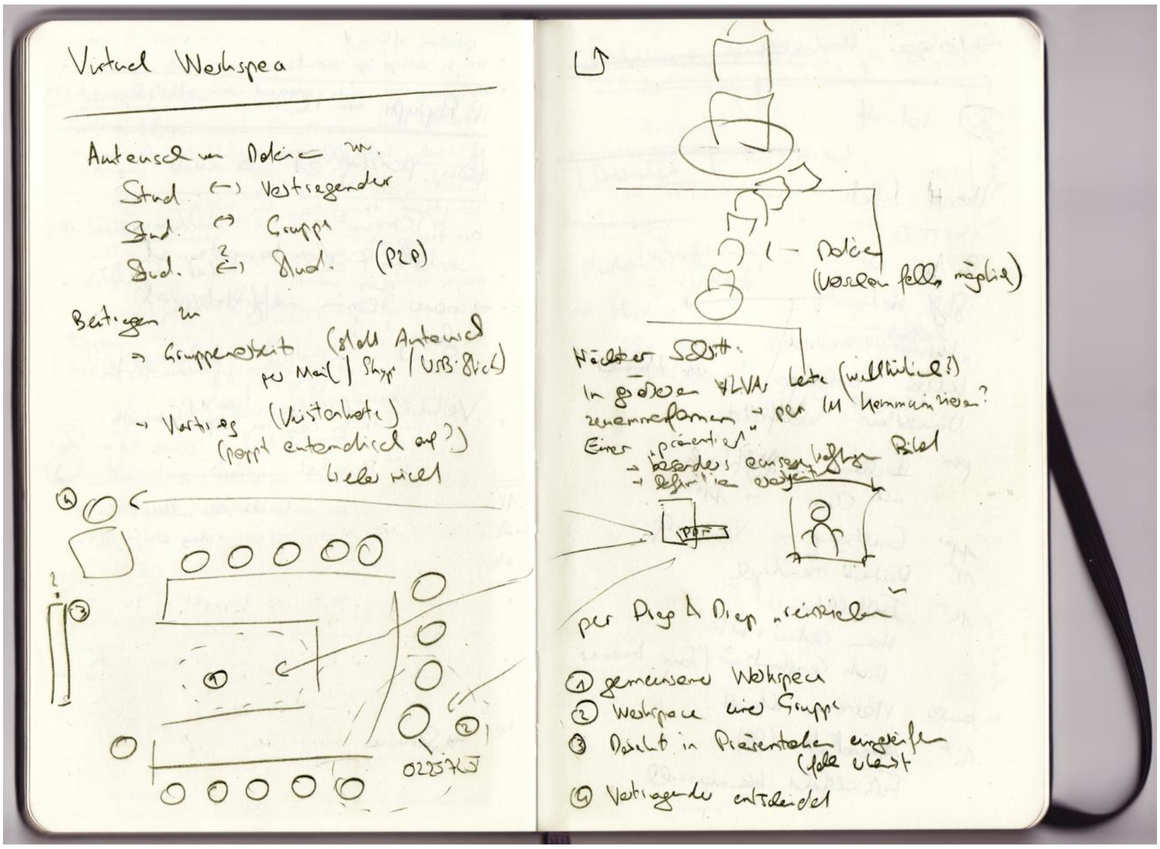

Figure 9: First sketch including room plan (bottom left), drag & drop operations (right page to left page), stacks of files (top right) and workflow (bottom right, steps marked in room plan).

Reflection

Several points of criticism can be made about this sketch. First, setup costs are very high in a new room. The lecturer has to draw a room plan and define the position of the shared area, the presentation screen, the tables and all of the seats. Since the setup of a room is subject to permanent change, the room plan might have to be adjusted every single time a lecturer is setting up the workspace. The students, on the other hand, have to find their seats. This is easy for a table with half of a dozen of seats, but becomes difficult when counting of rows and seats is necessary. When someone changes a seat in real life, the change also needs to be reflected in the application.

Second, file representations do not reflect the nature of the content to be exchanged. URLs like links to articles on Wikipedia or videos on YouTube are usually not transferred as a file. Also, group work might result in an idea, a definition or some other kind of text (or part of text). When the users are forced to use files in order to share and present them, they would be obliged to save this (maybe very short) text as a file.

Third, the application only knows two modes of sharing artefacts, since artefacts are either visible to the whole group or to a single other user. When the students work in groups of four, sharing artefacts can only be done one after another if they do not want to make their content publicly visible. Even if this is an option, the artefacts of all groups will be stacked on each other in an order that does not reflect the composition of the groups.

These deliberations led to the evolution of a different yet related concept. Instead of using the room plan to structure the content, the idea of a spatial structure came up, i.e. an infinite canvas where content could be arranged. But before designing such an application, it seemed important to explore the possibilities of this idea.

5.2 Developing scenarios and personas

Although the main impetus for the development of the new application came from a university setting, it became clear that it might be used as a universal tool for content sharing. In order to explore this task and discover new requirements, several scenarios and a few personas were developed. The scenarios and personas should broaden the view of possible usage.

Scenarios

In the context of software development, a scenario is "a representative example of user-system interaction to be used in system design" [31]. A scenario textually describes how a user interacts with an application. They aim to guide the development process and supplement functional system requirements. By imagining scenarios, designers and developers can focus on how a system will behave and how it will be used.

In the context of this work, scenarios serve as a way of illustrating the different contexts of use for the application. The scenario development was started with a brainstorming session; the result can be seen in Figure 10. Three of these scenarios were phrased into the following stories. The scenario aims at describing the overall setting, the general workflow while using the system and the content that is shared. Hence, the three following scenarios first describe the situation which the application is used in. Then, the workflow is imagined. Finally, the digital artefacts that were shared in the course of the workflow are listed.

Scenario 1: Event planning

Workflow description

A marketing agency is given the task of planning a birthday celebration for a well-known politician. Together with the representative of the politician's staff, location, room decoration, and the course of the events during the celebration have to be defined.

To explore the location options, the agency's project manager has prepared some suggestions and has put them into the workspace. The agency's project team and responsible person of the politician's staff gather in a meeting room to discuss the location. The project manager shows their suggestions while others add new possibilities to the workspace. The pros and cons of each of the suggestions are discussed and finally a decision is made.

For the decoration and the interior of the location, the project manager presents the floor plan taken from the proprietor's website. The position of the stage, tables, etc. is plotted on the floor plan with a dedicated software application. The screen of the project manager is cloned on the room's projector so that every participant has a good view of the current status. Ideas for furnishings are constantly published, presented on the screen and discussed by the team members.

The client's representative has prepared a rough sketch of the agenda in the form of a Word document. This is published, presented, discussed and completed. Some items on the agenda require additional investigation and consultation. It is agreed that smaller groups will agree on suggestions for some of the agenda points until the next meeting.

Two weeks later, the project team meets again and the subgroups present their findings. For the laudation, held by an honoured member of the party, important moments of the life of the celebrated politician were collected. A video is meant to be the emotional climax of the celebration. The subgroup has collected a collection of YouTube videos to show the direction. The subgroup concerned with the birthday cake has documented a dozen of photos to illustrate their ideas. All of these artefacts are presented on the screen, commented by the correspondent subgroup and discussed.

Analysis

In this scenario, text, Word documents, images, drawings, web videos and comments are shared via the application. Important to note is that the application does not provide features for formal meetings minutes or decision making.

Scenario 2: University course

Workflow description

In a university course about database systems with more than one hundred students in the lecture hall, an example is presented that entails a complex database query. The lecturer asks the student to come up with a solution, alone or in pairs. Then they ought to use their laptops, which are usually turned on anyway, to share their solutions. In the workspace, the lecturer has already published a database diagram which can be used as a reference. The students think about their solutions individually or in pairs and share them afterwards. The students are only able to see their own solution.

Ten minutes later, the lecturer brings the to-be-defined application to the screen and asks some students to show their solution. The bravest ones push their approaches to the screen and argue why it is a good one. As soon as they present their solution, it is also visible to all of the other students in the application. The lecturer comments on the solutions and initiates a discussion. He copies two of the solutions and pastes them in the query window of the database to compare their performance.

Analysis

Text and images are the only media types that are used in this scenario. Since the students do not have access to solutions that have not been presented yet, they are not able to copy other solutions. In a more traditional setting, sequential answers of students are of course influenced by each other. For the sake of learning, it might be useful to make the solutions visible the moment they are shared and let the students adapt their solutions.

Scenario 3: Design workshop

Workflow description

A graphic designer meets with a six member strong project group from her client for a kick-off meeting: A new corporate design is about to be established. The project group has collected all types of art work: Brochures, posters, websites, ads, door plates, letter paper, etc. They have brought along some of them in print, some in digital form. During the meeting, they discover further pieces of design that they download from their company network.

The digital ones of these artefacts are shared via the application. To clarify their point, one team member takes a photo of a minor detail of a brochure with a smartphone and shares it with the application. On the large screen in the meeting room, the digital artefacts are presented and discussed. Some of them are printed to be put on the walls of the meeting room for inspiration.

To collect more inspirational material, the graphic designer arranges the digital artefacts on a moodboard. It is complemented by pictures of products, employees, photos of landscapes and nature, to reflect the mood the corporate design should trigger. While the project group generates material, the graphic designer collects it and arranges it in a dedicated software application.

Analysis

A wide range of media types exist in this scenario: Text, images, photos, drawings, videos, websites, etc. Like in scenario 1, additional steps of documentation of the proceedings are necessary.

FURTHER SCENARIOS AND IDEAS

Obviously, the application is very helpful when a group of people works collaboratively on something that requires the use of digital artefacts like images, texts, videos or websites. The use of the application is not restricted to face-to-face-meetings: Given a synchronous channel like video and audio is also provided, the application extends the possibilities of a remote participant to share content with their colleagues.

Concerning asynchronous activities, it is imaginable that the participants share their content in advance in order to save time and to present it later in a face-to-face meeting. If the publisher is not available to present his/her content, a different set of requirements is brought up: Content has to be annotated, explained and commented on by the publisher as well as by other users. In a co-located situation, verbal communication is used to accomplish this activity.

To get an overview of the artefacts and activities that might appear, scenario 2 was expanded and described step by step in table form. In addition to the process, the resulting artefacts and the used software and hardware are also listed (Table 2).

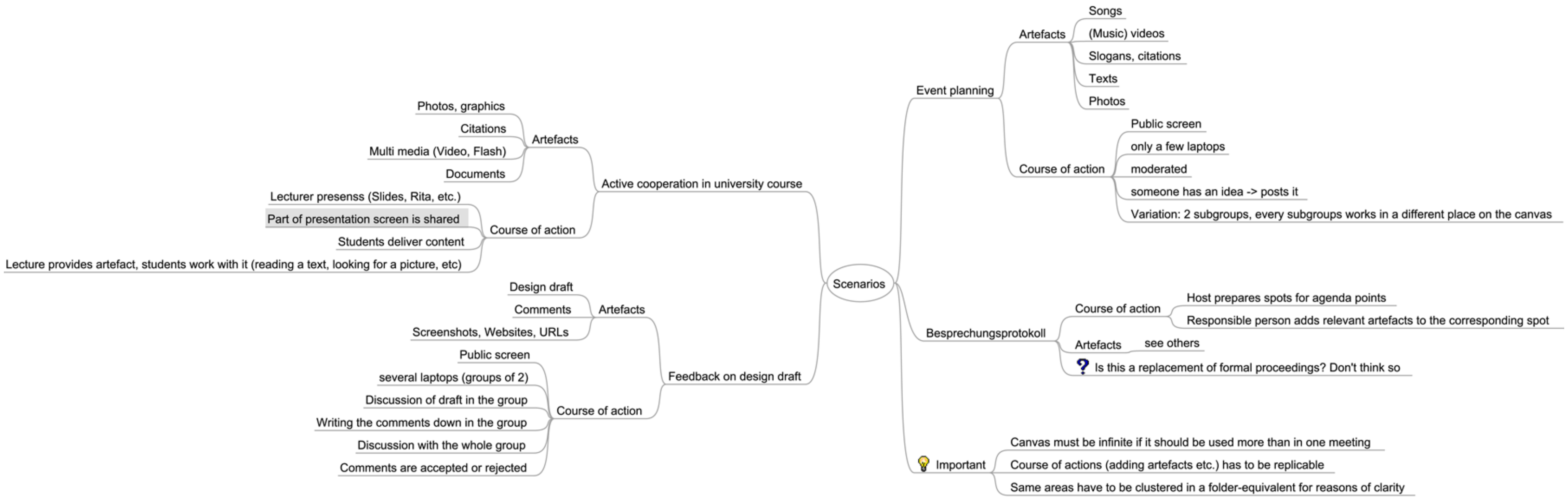

Figure 10: First scenario brainstorming

Description	Resulting artefacts	Software	Hardware
Start of university course			
Course instructor explains the guidelines for the course and the mode of completion.	Events Notes	Calendar software Text editing software	Laptop Smartphone Tablet computer
Course instructor starts his lecture about different design principles. The principles are discussed with the help of examples.	Photos Slides	Presentation software	Laptop
Some students talk about their experiences with the design principles and provide examples of their own work.	URLs Photos	Web browser File browser	Laptop Tablet computer

Examples of good design			
The instructor explains the homework: The student should look for a real object which can serve as an example of good design and bring it with them into class.	Notes	Text editing software	Laptop Smartphone Tablet computer
The students look for adequate objects and document their search: In the supermarket, online, in the streets, etc.	Photos Videos URLs	Web browser File browser	Mobile phone Photo camera Video camera
The students publish photos of the found objects prior to the next course.	Photos	Photo editing software	Computer Laptop
During class, the objects are presented and discussed. Further information is added to each object.	Photos URLs Videos Notes	Web browser Text editing software	Laptop Tablet computer

Seasons calendar			
The instructor explains the homework: The student ought to design a seasons calendar, i.e. a calendar with four sheets. Each sheet should use a different design principle.	Notes	Text editing software	Laptop Smartphone Tablet computer
The students sketch their seasons calendars and document their research.	Sketches URLs Photos Graphics Slides	Web browser Photo editing software	Laptop Tablet computer
During class, the students' calendars are presented and discussed. Further information and thoughts are added to each calendar or calendar sheet.	Photos URLs Notes	Web browser	Laptop Tablet computer

Poster design			
The instructor assigns the student to design poster about a specific colour. The students form groups of three and start collecting and sharing material.	Tasks Photos URLs	Web browser Photo editing software	Laptop Tablet computer

Create a presentation			
The students are assigned to create a presentation about an aspect of Web 2.0. They form small groups which will continue to work together during the next classes. The students of every group agree on how to split the preparation tasks among them.	Contacts Tasks Notes	Contact software Calendar software	Laptop Tablet computer Smartphone Mobile phone

All participants do research about their aspects of the subject and document their findings.	URLs Papers Citations	Web browser Library	Laptop Tablet computer Computer
In a Skype chat, the students exchange their latest findings.	Notes Chat protocol	Skype Text editing software	Computer Laptop Tablet computer Smartphone
A group member informs his/her group via mail about an interesting website.	E-Mail URL	E-Mail-Software Web browser	Computer Laptop Smartphone
A member of another group discovers an interesting talk and sends it to the group.	Slides	Web browser	Computer Laptop Tablet computer
During a lunch meeting, the current research status is discussed and a first draft of the presentation is assembled. Further work on the different parts of the presentations is to be carried out separately.	Events Tasks Notes Slides	Web browser Presentation software	Laptop Tablet computer Smartphone Mobile phone
The group members adapt their parts of the presentation and exchange it after each revision.	Slides	Presentation software	Computer Laptop
In a Skype conference, the final details are discussed and the slides are finalized.	Audio Slides Notes	Skype Presentation software	Computer Laptop Smartphone Tablet computer

Development of an audio application			
In a university course, an audio application is to be developed. The application should accept arbitrary data and convert it in an interesting audio presentation. In the first lecture, small groups are formed.	Tasks Notes Contacts	Calendar software Contact software	Laptop Tablet computer Smartphone Mobile phone
The students independently look for data sources that might be used for interpretation by the application.	URLs Data tables	Web browser File browser	Laptop Computer
In a Skype chat, the preliminary results and the next tasks are discussed.	Notes Chat protocol Tasks	Skype Text editing software Calendar software	Computer Laptop Tablet computer Smartphone
One student begins coding the application.	Software code	IDE	Computer Laptop
Another student is responsible for preparing the data. For this task, characters have to be extracted from an image.	Images	Web browser OCR software	Computer Laptop
In a final Skype conference, the data and the application are finalized and a presentation is outlined.	Audio Notes Slides	Skype Presentation software	Computer Laptop Smartphone Tablet computer

Analyze an ATM			
The students are assigned to analyze an ATM and derive recommendations for improvements based on their findings.	Tasks		
The students observe people using an ATM: How long does the transaction take? What does the interface look like? Etc.	Notes Audio Video		Smartphone Photo camera

Table 2: Scenario 2, described in great detail

Biographical data	Goals	Example scenario
Karin Engleitner Student, 23 years, in a relationship	• Get started fast • Get finished fast • Learn something on the way • Have enough free time	Creating a collage "Children & free time" in a group of four • Brainstorming, a mind map is created. • Tasks are assigned, every group member is responsible for one arm of the mind map. • Every group member posts pictures, URLs etc. related to the words. • Review with the whole group.
Patrick Schönkerner Graphics designer, 28 years, married	• Deliver a good result to his customers • Collect inspirational objects • Work professionally	Collecting ideas for a newspaper relaunch • Feelings that should be evoked by the new layout are collected. • The old layout is commented on. • The first drafts of the new layout are commented on.
Katharina Seidl Event planner, 45 years, divorced	• Keep her company in business • Surprise customers with creative and inventive ideas	Planning a birthday party of a celebrity • Agenda of the event is sketched. • Pictures from the last party are posted.

	• Make planning an event an experience itself	• An idea for the design of the birthday cake is sketched. • Ideas for artists that should perform are collected.
Wolfgang Reis Assistant lecturer, 44 years, married	• Avoid to have bored students in his class • Provide students with knowledge in a way that they remember it • Involve the students in the teaching	Lecture “Universal principles of design” • Preparation: A lot of examples of different design principles are posted. • Students are assigned to find a screenshot for every principle. • Students collect and publish screenshots. • Screenshots are discussed in the lecture.

Table 3: Personas

Personas

To gain more insight in the users of these scenarios, four personas were created, as shown in Table 3. For each persona, the most important goals were phrased in a few words. Additionally, the most interesting scenario to this persona was outlined. The personas were given a name and some biographical data. They only served as another form of exploring how the desired application could be used.

Conclusion

The development of scenarios and personas showed that there are a lot of applications for a tool that supports sharing and publishing digital artefacts in and beyond meetings. Of course, the university setting was explored in greatest detail, but other applications are very likely. It was shown that a number of other users like media agencies, designers, teachers and event planners experience scenarios where sharing und publishing digital content as a group is part of their daily work. Thus, the work on designing a tool supporting these scenarios was continued.

5.3 Virtual Workspace

The first sketch used a digital representation of the real room as the prime structure to organize content. In addition to the high setup costs that were described before, this has another serious drawback: Artefacts connected to the same idea or subject are spread over the whole workspace since they are displayed as stacks on the seats or the common area. The artefact's meaning and relation to other artefacts as regards content is not reflected in the presentation of the artefacts.

Starting with this initial point, the concept called *Virtual Workspace* was developed. Basically, it consists of a persistent, collaborative and infinite canvas. The idea of an infinite canvas originates from an application named Rita [32], a Mac OS X software for a single user that supports an infinite instead of a fixed image size.

In *Virtual Workspace*, the artefacts can be dropped on the canvas. The artefact will then be displayed by a rectangle showing its content or a preview of its content. The users can resize and arrange the artefacts so that their position reflects the content-specific relation to each other. All users are allowed to carry out these actions, since it is assumed that social interaction will be used to agree on a fitting size and position.

Sharing artefacts on the canvas is possible in two ways: Like in the first sketch, the user can drag and drop a file from their computer to the application. The file is then uploaded and visible on the canvas. Alternatively, content from the operating system's clipboard can be pasted on the canvas. In addition to that, a toolbar allows the creation of a paragraph of text, a freehand sketch and annotations to an artefact. These annotations are thought to serve discussion and documentation purposes. Apart from the text itself, an annotation contains the user's name and a timestamp.

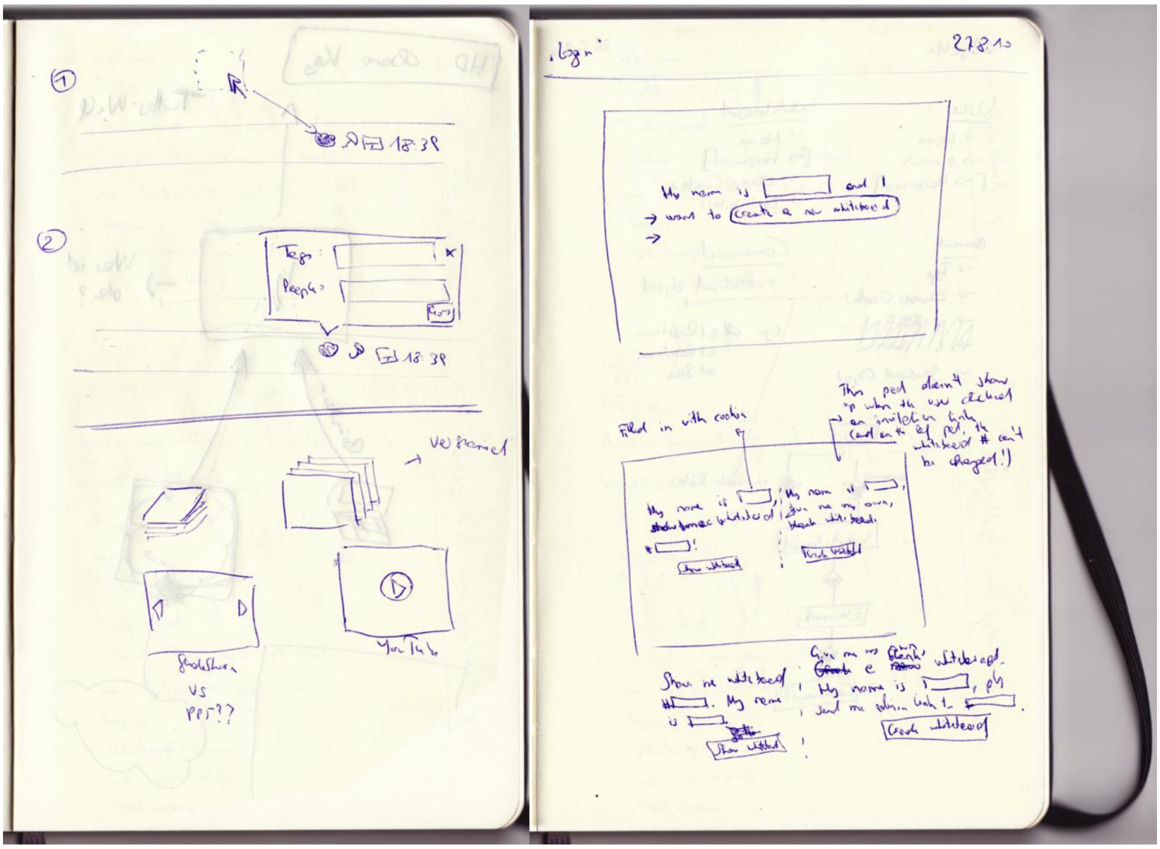

Figure 11: Dropping a file from the desktop (top left), different content types (bottom left), login forms (right)

Virtual Workspace not only displays a preview of the artefact's content, it also allows interaction with it (Figure 11). From the scenarios, it was clear that a broad range of content will occur during the work with *Virtual Workspace*. Different types of media require different presentation formats; some of these are listed here:

- Image data is displayed in a picture frame. In addition to resizing and moving, rotating is allowed.
- If several images are dropped on the canvas at the same time, these are displayed as a stack of pictures with symbols for browsing between them.
- Video data is displayed in a video player frame.
- Documents like PDFs or DOCs are displayed in a frame with a ratio of a sheet of paper and a scrollbar or symbols for paging.
- URLs are generally displayed in a browser-like frame on the canvas. However, for specific domains like youtube.com or slideshare.com the respective player API is used. In this way, videos or presentations can be viewed and interacted with directly on the canvas.

For presentation purposes, the part of the canvas that is currently being displayed on the presentation screen is marked as a rectangle on the user's view of the canvas. This rectangle can be moved and resized to define the view of the presentation screen.

All of the artefacts are publicly visible, private exchange of files from one user to another is not supported. Different groups however, can place the content that they are working on somewhere on the canvas where no other group is working. By placing them at a distance from the presentation screen's view and the work of others, it becomes obvious that these artefacts are not meant to be presented at the moment.

In all scenarios, the application itself is not in the focus of the users. It is used as a medium to accomplish a goal, i.e. generate a new idea or collect inspirational material. Thus, it is important for the application to integrate seamlessly in the group's workflow on the one hand and in the user's workflow on his/her personal device on the other.

However, users might want to know where their colleagues are adding new artefacts. This is especially important in scenarios where the participants work in subgroups in different places on the canvas. These actions are indicated at the edge of the application window. When another user adds an artefact somewhere on the canvas, an indicator including the name of the contributing user and an icon of the content type appears at the edge of the screen. The position of this indicator shows the direction which the user has to scroll to in order to see it. When the indicator is clicked, the workspace is scrolled to the position of the new artefact. Most of these interactions can be seen in Figure 12.

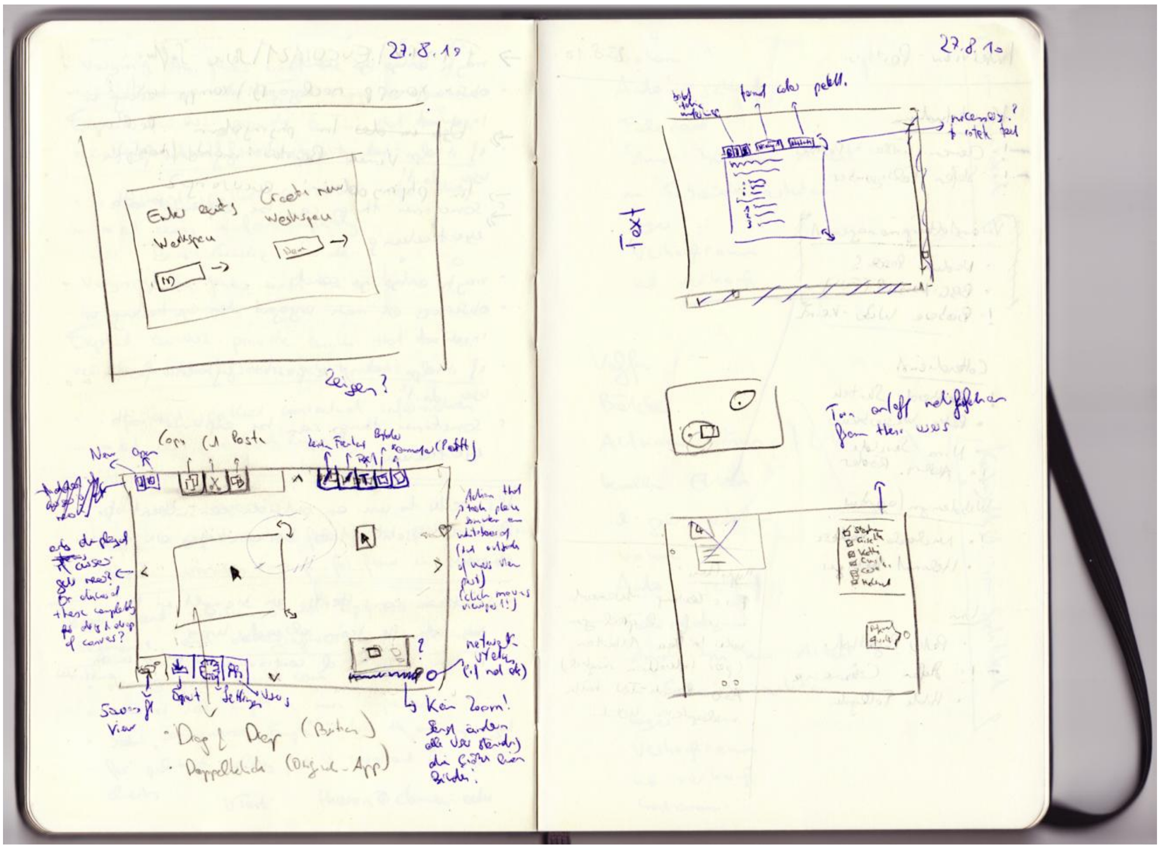

Figure 12: Sketch showing login screen (top left), interactions on the canvas (bottom left), text editing (top right), selecting an area for display on presentation screen (middle right), notifications (bottom right).

Prototype

To explore the possibilities of such a concept, a prototype was developed. It was clear that drag & drop as well as copy & paste operations must be possible in order not to force the users into cumbersome workarounds like saving a short text or a part of an image into a file instead of pasting it.

Since the installation of a dedicated software raises the barrier of entry for the users, the prototype should run in a web browser. Unfortunately, operations like drag & drop and copy & paste are not supported in all modern browsers: The W3C's draft for HTML5 defines drag and drop operations for files [33], but these features have not been widely implemented yet. Drag & drop of file segments (i.e. the selection of a picture) or clipboard support for non-text formats is neither specified nor implemented.

This was the reason why the prototype was implemented using *Adobe AIR* [34], a runtime environment by Adobe for web applications using HTML, JavaScript, Adobe Flash, and ActionScript. Developers can use web technologies to create applications that run natively on Windows, Mac OS and Linux. This framework was chosen because it adds a JavaScript API for drag & drop and operations with the operating system's clipboard. However, it has one considerable disadvantage: The AIR environment has to be installed, also the application itself. This raises the barrier of entry; a trade-off that seemed negligible given the added possibilities of interaction with the operating system.

When the prototype is started, it presents an empty screen. There are neither a toolbar nor a menu. The user can drop a file or paste clipboard data on the canvas (Figure 13). The client uploads the artefact to a web server via an internet connection. The client checks for new artefacts on the server every few seconds. If a new artefact is found, it is downloaded, positioned, resized and finally displayed on the canvas. Artefacts whose size or position has been changed by another user are updated accordingly.

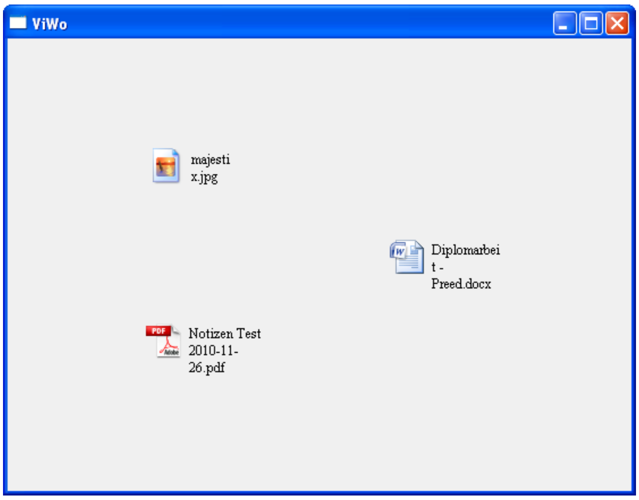

Figure 13: Prototype of *Virtual Workspace* with three artefacts in the workspace.

REFLECTION

Virtual Workspace describes a collaboration studio rather than a simple tool to share and present artefacts. It offers a wide range of functions, which makes it suitable for a likewise wide range of situations, extending the starting presumption of a co-located meeting to the management of a whole workflow which takes place in different locations and at different points in time.

Since the scope of the idea got bigger, it was decided to take a step back and do more research to answer the question if the product idea was worth realizing. Interviews with prospective users should help in gaining more knowledge about their work routine and how a tool like *Virtual Workspace* could support it.

5.4 User research

As shown in chapter 4, interviews are an important part of ethnographic research which aims at understanding how people behave and work in a certain context and why they do this. To gain more insight into the work routine of prospective users, three people were chosen. All have in common that they are doing a study closely related to design at the moment or that they have completed such a study recently. All of them work in this field, fulltime or part-time because of an ongoing study. This choice was made on purpose to have opinions of people with a shared background but different fields of work. A broader selection of interview partners might have led to such a diverging range of opinions, work realities and applications that it might not have been possible to derive reasonable insights.

All interview partners were told that the interview will cover their experiences in working with digital artefacts in work or university settings. The idea of an infinite canvas was not unveiled in advance. Some open questions were prepared by the interviewer to get the conversation started and to act as a kind of checklist to assure that the most important aspects of the interview partner's work routine were discussed during the interview. To avoid self-censoring, all interviewees were ensured that their names and the names of their employers would not be unveiled. The following names are therefore pseudonyms made up by the author. First, the three interviewees will be introduced and then several topics of interest will be presented.

Gregor is 27 years old and studied "Graphics and Advertising" at the University of Applied Arts in Vienna. He founded a graphics and advertising studio together with a colleague from university. They design editorials, layouts, logos, illustration, fonts and make stop motions films as well as soundtracks.

Florian's age is 28, he earned his degree in media informatics at the Vienna University of Technology and continued his studies at the University of Applied Arts Vienna in the master programme "Digital Art". He is employed in a design agency with a focus on interactive installations in museums and foyers of large companies.

Christoph is 23 years old and studies media informatics at the Vienna University of Technology. Besides his studies, he works at a SEO company. He is responsible for an internal software application which is used by his colleagues to keep track of certain URLs.

When asked about group work in university, the students of the University of Technology moaned. Group work was seen as a chicane for the students, since it meant finding qualified colleagues, doing a lot of coordination and discussing. Groups typically consisted of three or four students, maximum six. Usually, the students only met once, at the beginning of the course. The amount of work was divided and deadlines were agreed on. In bigger groups, pairs of students met from time to time in the canteen or online via Skype. Files were exchanged via mail or in Skype group chats. Christoph remembered writing a text collaboratively with *Piratepad* [35], a web-based application for collaborative writing in real time. When communicating with students outside of the group, different message boards were used.

Interestingly, students of arts were never given a task that had to be done in a group. Occasionally, work was assigned to pairs, but hardly ever to a group of three or more students. In weekly review meetings with the whole class, consisting of 30 students, the drafts were presented to the class and the professor. However, mostly only the professor commented on the work. When asked, Florian and Gregor stated that it is uncommon to leave e.g. a poster in the classroom as a kind of inspiration for others. Both agreed that collaborative work is not fostered at the university.

Concerning lectures and tutorials, the three interviewees said that very few professors emphasize contributions by the students during the lecture. Some lecturers might ask questions, but they could not remember bringing material to the lecture or doing online research during the lecture. Florian und Christoph, the two students at the University of Technology, remembered only a single exception.

In Gregor's design studio, Gregor and his colleague usually work on different projects. The interview took place in their office which is situated on the self-renovated first floor of an old backyard. Although they request feedback from each other, they don't share digital artefacts with each other. Communication with the customers is mostly done via mail or phone. Website designs are sometimes done without a single meeting.

Florian also works on his project alone. When he thinks an idea is developed far enough, he presents it to his boss and later to the client. Some clients have very detailed ideas about the resulting installation. If there is still room for ideation, a brainstorming is usually done without a client's representative.

Stefan's work is organized in an Excel sheet called "Excel sheet of hell". Since he is not in contact with clients, he only reflects new ideas with his colleagues. New features for the software application which he is responsible for are documented in the Excel sheet. He would like to use *Cacoo* [36], a web-based collaborative interface design tool. The company follows the Scrum development model which includes daily morning meetings. These meetings are logged on paper or on a shared whiteboard. More complex ideas are sketched on the whiteboard while his colleagues stand around it. However, these ideas only embrace minor improvements. Solutions or feature requests of a bigger scope usually come directly from the company's boss.

When presented the idea of a persistent, infinite, collaborative canvas, all of the interview partners found the idea appealing. However, they could not think of a single situation in the last week where the application might have been useful for them. Most of their work is done alone and brainstorming does usually not include digital artefacts but more hand drawn sketches or words.

FINDINGS IN USER RESEARCH

Reflecting the scenarios, a tool like *Virtual Workspace* could be used in a wide variety of situations. The range of scenarios shows that *Virtual Workspace* could be used with groups of different size, co-located as well as dislocated, at the same time as well as at different times. There are few tools for brainstorming with digital artefacts, so *Virtual Workspace* should fill this need.

However, when talking to possible users it was discovered that they were challenged by finding a situation of their daily work routine where this idea of a persistent, collaborative and infinite canvas with a strong spatial component might be useful. Although they liked the idea, they could only think of theoretical situations where it might be appropriate to use it. Surprisingly, it was also discovered that the interview partners almost never had to work with digital artefacts in a collaborative setting.

To the author, there are two possible explanations for this finding. First, that the idea of such a tool as a whole might be a bad one and that there might be no field of application for such a tool. This seems unlikely, given the wide range of scenarios, but cannot be neglected. The second explanation for the users not being able to find a corresponding situation in real life is that *Virtual Workspace* breaks too many very common paradigms. This makes it very hard for prospective users to think of it in their daily work routine. Files for example are usually displayed in a list or a grid, sorted by criteria closely related to it like date of creation or file size. Closely related files are organized in folders, not in a spatial way although this is very common with sheets of paper on a desk or books on a shelf. Also, software that is able to interact with different types of content is very rare. Only some very invisible and small tools like an operating system's preview application work with a variety of file formats, all others are specialized on a few.

Assuming that the concept is not a bad idea as such but given that it is very uncommon to use, the development of such a tool might turn out a successful game changer with users who rid themselves of their old habits. Or as a complete failure since the users find it too cumbersome to adapt their usage patterns.

Recalling that the tool will be used only a few times a month during a workshop, breaking these usage patterns also raises the barrier of entry. Consequently, another paradigm has to be found.

5.5 Preed

The two previous sketches respectively prototypes used spatial information to organize the artefacts: Either on a room plan or on a two-dimensional canvas. However, there is another interesting criterion to structure content: Time and sequence. It is important to note that only the presentation and structure of the content were again subject to discussion. It was clear that the interactions with the content like zooming into an image or playing a video described in the previous section were taken for granted.

This time, also the start of the application has been considered. In contrast to the first sketches of *Virtual Workspace*, the interface is even more reduced. When the presenter starts the application, only a single field is displayed where a name for the workspace can be configured. The users access the workspace via a URL consisting of the application's URL and the name of workspace, e.g. http://feed.bazalka.at/workshop3, where "workshop3" is the name of the workspace. Users are only prompted a (nick)name, which is later displayed next to the artefacts that they have posted.

But how should the content be presented? A look at the two popular Web 2.0 services Twitter and Facebook shows that both use a very similar mode to present the users' content: In a vertical list, beginning with the artefact that was posted most recently (Figure 14). Based on Twitter is a concept that is widespread in workshops and presentation at conferences: A Twitter wall (Figure 15). Via the Twitter API, messages on Twitter ("tweets") containing a special word ("hashtag") are collected and displayed on a public screen. This is meant to support feedback to the presenter and encourage discussion among the participants.

Figure 14: News feeds of Twitter (left) and Facebook (right, blurred for privacy reasons)

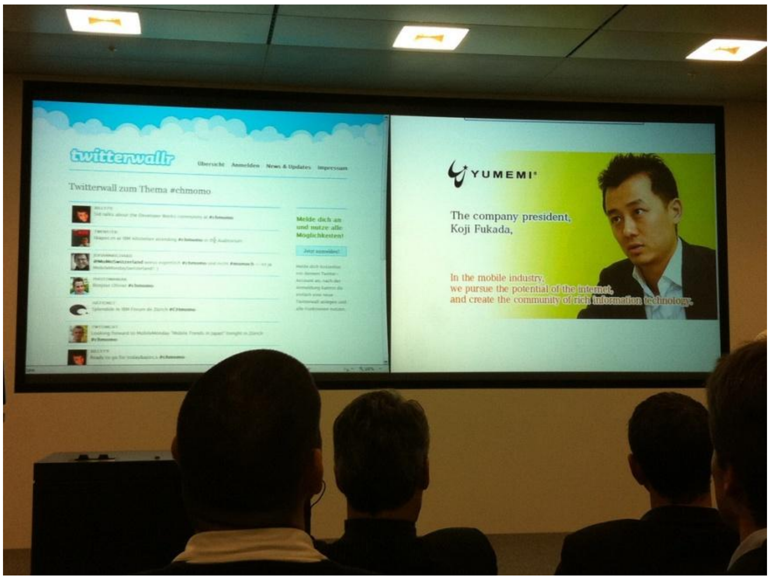

Figure 15: Twitter wall at a conference [37]

Beginning with the Twitter wall in mind, a sketch outlined an overlay to the lecturer's presentation: Users can publish content via their web browser. On the public screen, the presentation of the lecturer is shown. Small icons are displayed at the edge of the screen as an overlay. Hovering this icon with the mouse displays the whole artefact (Figure 16).

To realize this idea, several options were considered. Technically speaking, there are input signals and one output signal. A signal from the presenter's device and a signal from another device delivering the content feed are merged into one output signal, which is shown on the presentation screen. One option is the use of a video keyer. This piece of hardware accepts two or more video sources, processes them and outputs one merged video signal. The composition can be based either on making a previously defined colour transparent ("colour keying") or on using an alpha channel which controls transparency in one of the input channels. Both techniques are common in a professional context. Colour keying however can pose a problem in the context of *Preed*. Of course, the web application could use a special colour to indicate areas of the video output that should be transparent in the output signal. But nothing prevents a user from posting content that also contains this colour. This might lead to very strange effects in the output signal. When a video using the special colour is published, every pixel with this colour in the video will be made transparent and the presenter's content will shine through. For a reliable setup, this is an inacceptable risk. The alternative, i.e. using an alpha channel to create transparency, is even more difficult. For every pixel, only values for red, green and blue are transmitted. An alpha channel needs to transmit a forth value indicating the level of transparency for every pixel. An additional greyscale video signal, synchronized with the cloud's content, could act as a mask and make certain areas transparent. Adding an alpha channel with consumer equipment is hardly possible.

Additionally, this solution does not allow an interaction with the published content on the presenter's device. Of course, hovering an item with the mouse is not possible. To provide this interaction, it would be necessary that some kind of software is installed on the presenter's device. This software would need to create the earlier described overlay and react to mouse actions. However, software installation is not an option, as described earlier.

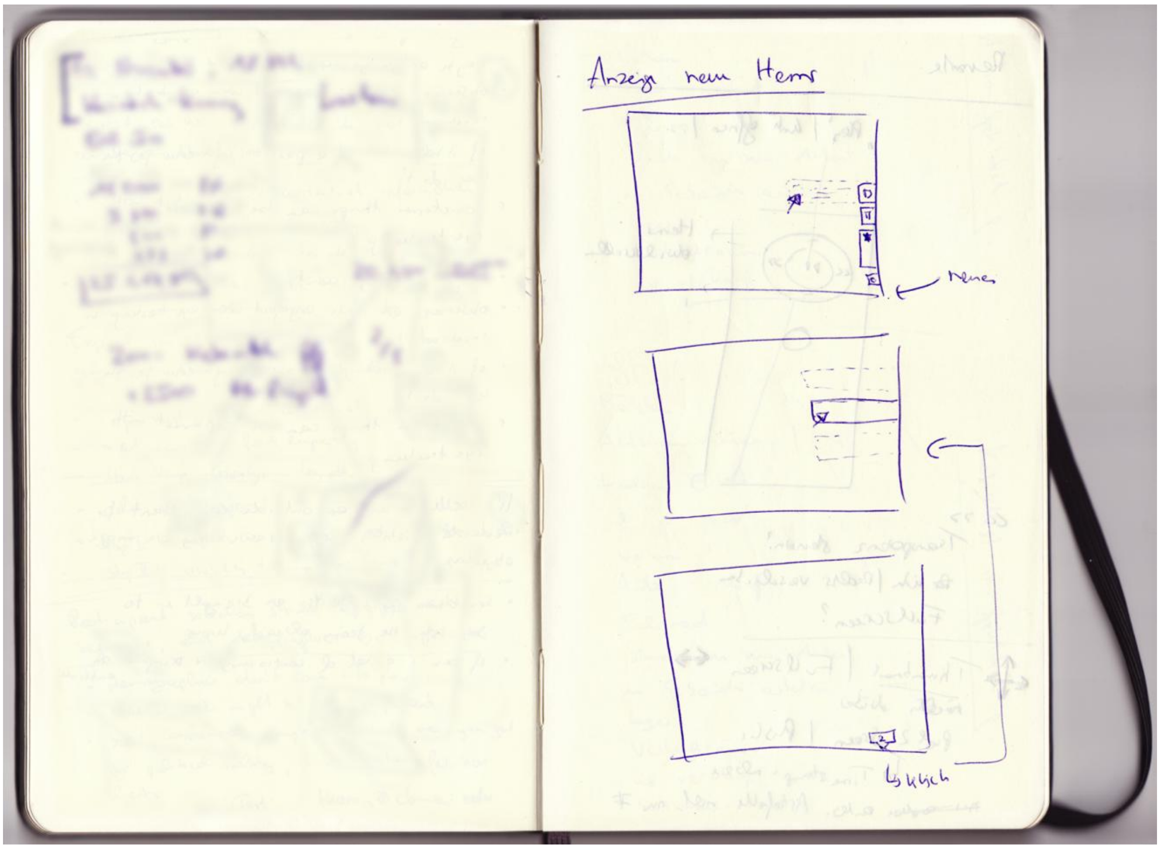

Figure 16: Different possibilities for displaying content as an overlay to the current presentation. (Left side blurred for privacy reasons.)

While this problem remains unsolved, another issue has to be considered. The overlay might interfere with the lecturer's presentation and pose a distraction to its content. Therefore, a different model was explored.

By default, only the presentation is visible on the public screen. The presenter has an Apple Remote (Figure 19) to control the application. A button switches between the presentation and a view of the artefacts that the participants have posted. Using a directional pad on the remote, the presenter can select an artefact and show it full screen with another button click on the remote. With another two buttons on the remote control, the presenter is also able to go back and forward through the list of artefacts in full screen mode. Figure 17 shows how the Apple Remote is used to interact with the application; in Figure 18 the grid view of all artefacts is sketched.

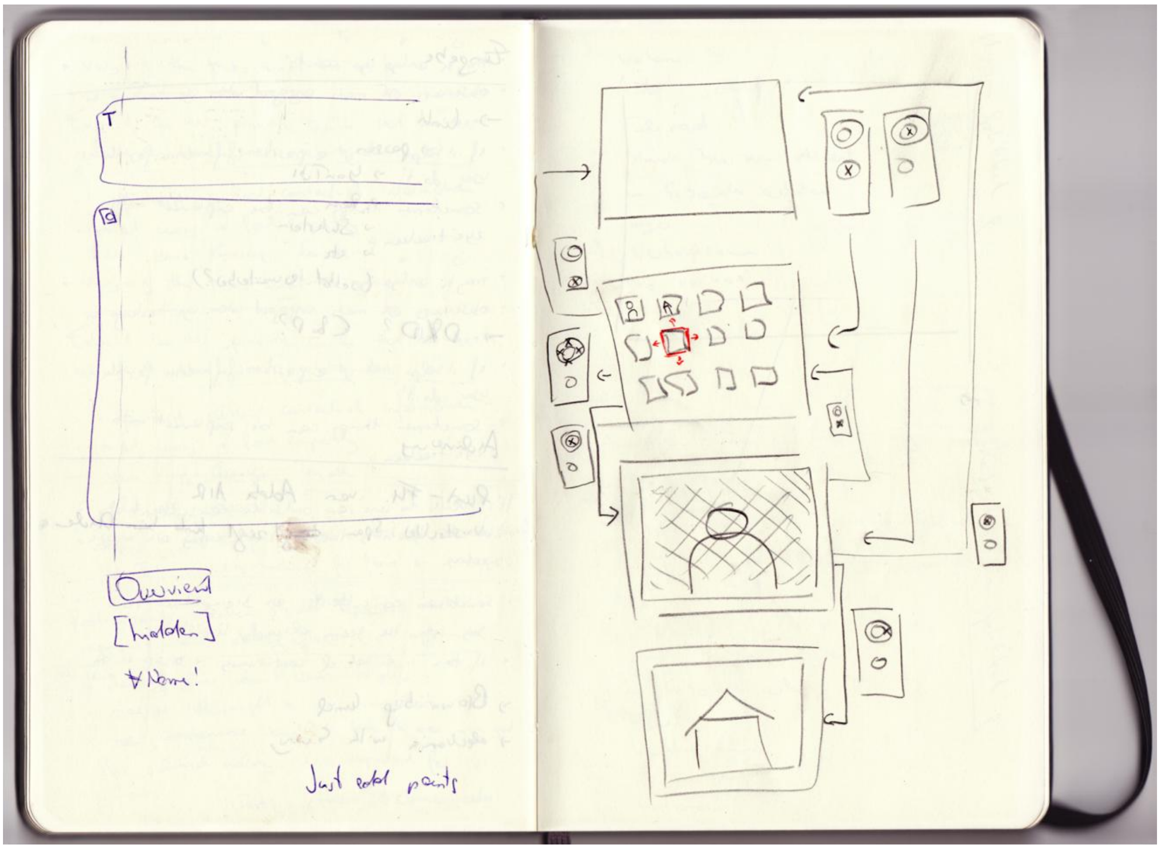

Figure 17: More possibilities for displaying content as an overlay (left), interactions with the remote, switching between transparent, grid and full screen view (right).

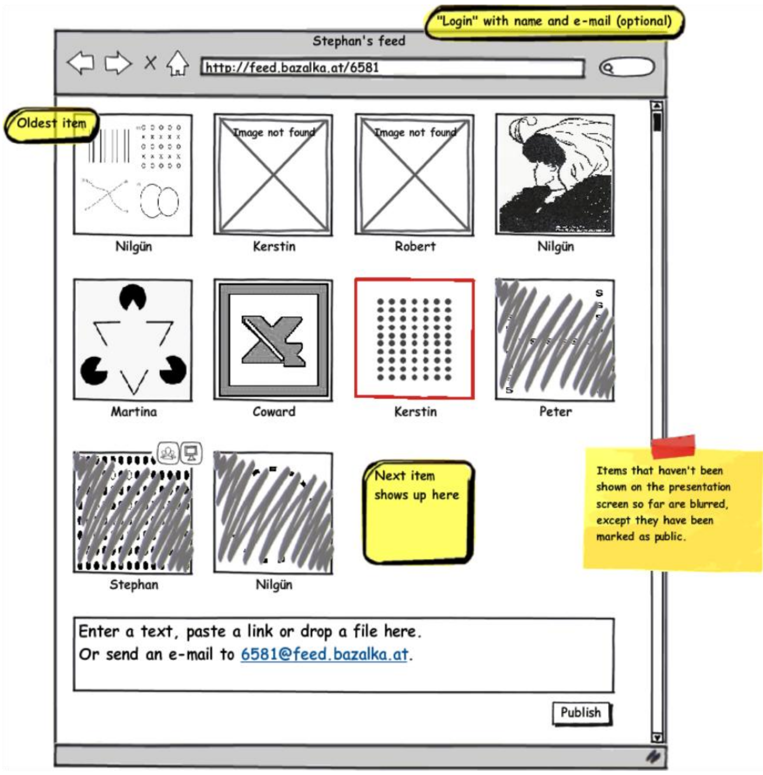

Figure 18: Contributor's interface, displaying all artefacts in a grid view and a publishing form.

It is assumed that the public overview of the artefacts is very similar to the interface the participants see in their browser. Only the form for publishing a new artefact is added, as well as icons for sharing it with the group and pushing it onto the public screen.

Differing from the first sketch, the artefacts are presented in a grid view. In the users' web interface items that were already shown on the public screen (either pushed by the publisher or selected by the presenter) are shown with a preview, new items are greyed out. The grid view offers a quick overview about past artefacts and an easy selection of a certain artefact.

However, many questions remain open: The presenter might want to preview the artefacts. Usually, the presenter's screen will show his presentation, so a further device would be necessary in addition to the remote control. Previews of all artefacts could be shown on the public screen, but in this case greying out the new ones in the user's web interface is rather useless.

In addition to that, it has to be questioned if such overview is necessary at all. Most of the time, a certain artefact will be shown on the public screen, making the private web interface insignificant at this moment. Before a certain artefact – be it one of the user himself or of another participant – is pushed to the public screen, it will most probably be inspected in full screen mode. Consequently, the grid view is not a necessary feature.

Furthermore, when sketching the interaction with the Apple Remote, it was discovered that six buttons are enough to select an artefact, but there is no button left to control the artefact. Certainly, the presenter wants to start a video, scroll through a website or zoom into an image with the remote.

Considering these arguments and recalling the structure of a Twitter wall, another sketch tries to cope with these challenges.

On the left hand side of the screen, a list of all artefacts is shown. New artefacts are added at the bottom of the list, the currently shown artefact is marked. For every artefact, the (nick) name of the publisher and an icon is shown. In case of a URL, the icon is the favicon of the domain. For files, the icon indicates the media type, e.g. image, text, video, etc. Additionally, another icon offers the possibility to push an artefact to the presentation screen. On the remaining screen area, the artefacts are displayed full screen, one by one. They are mounted vertically on an infinite plane. When the user selects an artefact in the list, the plane quickly scrolls to this artefact.

At the bottom of the screen resides a text field to publish a new artefact. Text can be typed in or a URL can be pasted. Alternatively, the users can upload a file from their device to the application by clicking "upload a file" and selecting the file with a standard dialogue window.

The presenter uses a Wiimote (Figure 19), a remote included with the *Nintendo Wii*, to control the application in presentation mode. Pressing the A or B button of the Wiimote lets the plane scroll to the next or previous artefact respectively and marks the artefact in the list accordingly. The directional pad is used to scroll a website, the buttons labelled with "+", "-" and "home" serve as controls for zooming into or out of an image or resetting it to its original size. With "1" and "2", the presenter switches between his presentation and the artefacts view.

Figure 19: Apple Remote (left) versus Wiimote (right, button B is on the back of the remote)

There are only slight differences between the client's and the presentation interface. The latter has no publishing form and does not show the icons to push an artefact to the presentation screen. Both functions are not necessary in the context of the presentation screen.

For the first time, this conceptual design seemed to be sophisticated enough to test an application with real users. A software prototype was developed using PHP as a server and HTML/CSS/JavaScript as client technology. Also, a hardware setup was developed to meet the requirements.

5.6 Exploring a hardware setup for Preed

Several hardware solutions were considered. This section will give an overview of the possible and actually used technologies. To understand the selected solutions, it is important to first shed light on the fundamental requirements for this application.

Preed wants to overlay content from the presenter's device (most probably a laptop) seamlessly with content from other users. Since *Preed* should not need any software installation, it is obvious that some kind of hardware setup is required for layering these two sources of content. Still, *Preed* has to be capable of working in full HD, meaning layering of images that are 1920 pixels wide and 1080 high. This has to happen continuously, i.e. a change on the presenter's device has to lead to an immediate change on the presentation screen. For slides of a presentation, a screen refresh rate of 2 Hz (twice a second) is enough. But the presenter might as well want to show a video, type a text or draw a sketch on his device. Since the use of *Preed* must not lead to any disadvantages in presenting content, a minimum refresh rate of 30 Hz is necessary.

Similar considerations apply to content which is published by other users. This content has to be displayed in full HD, without delay and again at a high refresh rate. However, this content is not generated by a single device, but by several devices across the room. These devices contribute to the content cloud. As described earlier, the output of the content cloud is controlled with a remote control.

Technically speaking, there is one output signal and two sources of input, i.e. the signal of the presenter's device and the signal of the content cloud. These two devices need to be connected with another device which switches between the two input signals and delivers one output signal to the presentation screen. Figure 20 shows an illustration of this scheme.

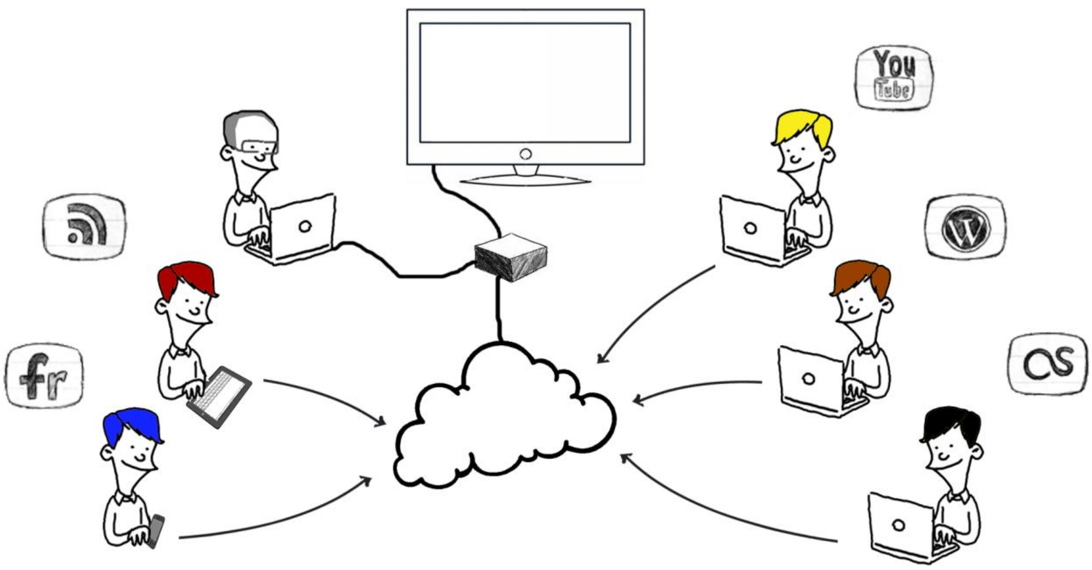

Figure 20: Users post content to the cloud, a black box switches between the cloud's signal and the one of the presenter's laptop. The remote control is not shown.

Thus, there are four players in this game:

- The presenter's laptop delivers a signal containing the presentation content
- The cloud delivers a signal containing the participants' content
- A black box switches between these two signals
- A remote control which controls the black box on the one hand and what cloud content is presented on the other hand.

All of these players will be discussed in the next sections.

PRESENTER'S DEVICE

For delivering a presentation with digital content, two mechanisms are widespread: First, the meeting room facilitates a computer that the presenter can transfer his presentation to. Second, the presenter brings a portable device, mostly a laptop, with him/her and connects this device to the presentation screen or projector using a common interface like VGA or DVI.

Both options have their drawbacks. The first one poses the risk that the computer in the meeting room is not able to display the content. The presentation software might not be present in the same version or a video codec might not be installed. This setup also strictly limits the presenter to the content that s/he has transferred to the room's computer at the beginning of the presentation. It is not possible to spontaneously switch to content of another presentation, earlier work or a website. – The second option means that the presenter has to have a portable device with him/her. This device also has to provide an interface which is compatible to the interfaces in the meeting room. Also, the device has to provide the presentation screen or projector with a signal with an adequate resolution.

Since the target resolution of *Preed* is full HD, it was assumed that the device supports this standard. This is true for laptops with an HDMI interface, including Apple MacBooks with a (Mini-) DisplayPort- or Mini-DVI-to-HDMI-adapter.

CLOUD

The cloud in Figure 20 has to provide an interface for the users to publish their content. The users on the other hand need a possibility to connect to the cloud. Since *Preed* should not require any software installation, an easy and comfortable way to satisfy these needs is a web application running a web server accessible via an internet connection. In a university as well as a conference setting, it is common to provide an internet connection to the students or participants respectively.

The web application has two interfaces which were described in the previous sections in great detail: The first one gives the users the ability to publish artefacts and push them to the presentation screen. The second interface is displayed on the presentation screen, i.e. the last artefact that was pushed by a user or an overview of the artefacts. To deliver this interface to the black box which switches between this and presenter's content, a device running a web browser with the web application in full screen mode is needed. In the reference setup, a Mac Pro running Apple's web browser Safari was used.

BLACK BOX

This part of the setup is the most difficult one. It has to accept the output of the presenter's device and the output of the cloud as input and process these sources. In the context of this work, two options were considered, which will be presented in this section.

First, an HDMI switch could be used that would control which of the two input signals is relayed to the presentation screen. However, by nature of this procedure, the output signal to the presentation screen is lost for a short period of time. Many HDMI screens need some seconds before they display a new signal, because they need to determine the exact resolution and frame rate of the new source. Of course, the possibility to switch between both sources quickly is an important factor for the acceptance of the setup, so this approach was also dropped.

The second option proved to be the most promising. A computer able to capture the output of the presenter's device in real time at an acceptable frame rate could display this content in a full screen window on its own screen. In another full screen window on the same computer a web browser is running the *Preed* web application. Following the keystrokes on the remote control, a script of the operating system brings the one or the other window to the front.

Inexpensive HDMI capturing devices were rare at the time of research, but a suitable one was found. *Blackmagic* offers a PCIe-Card named *Intensity Pro* [38]. It has an HDMI in and an HDMI out interface and comes with *QuickTime* compatible drivers. When plugged into a Mac Pro, a simple patch in *Quartz Composer*, a node-based visual programming language, is sufficient for displaying the captured video input full screen on the Mac Pro's monitor.

Unfortunately, the *Intensity Pro* is not very flexible regarding the variety of input formats. The resolution and frame rate have to match common TV standards exactly, meaning 1920x1080 pixels wide images, 24 or 30 images per second. Some devices are only able to deliver full HD resolution at a lower frame rate, which renders them incompatible with the *Intensity Pro*. Apart from that, the card has a very low latency and captures content uncompressed.

Summing up, a Mac Pro turned out to be the black box. The presenter's device is connected to the HDMI in interface of the *Intensity Pro*. This live image is displayed in a full screen window with the help of *Quartz Composer*. The Mac Pro's monitor output is delivered to the presentation screen, meaning that it looks as if the presenter's device were connected to the presentation screen directly. In addition to the first full screen window, Apple's web browser *Safari* is also running on the Mac Pro and showing the presentation interface of *Preed*, i.e. the content shared by other users. The only remaining question is how to determine which of these two windows will be in front and visible to the participants at a certain point in time.

Remote

Using a Wiimote with Mac OS X is rather easy. The remote transmits keystrokes and sensor data via Bluetooth. Given the Mac Pro is equipped with a Bluetooth receiver, software like *OSCulator* can process keystrokes of the Wiimote and start certain actions on the Mac Pro. In the context of *Preed*, a short script written in AppleScript is called for every keystroke. The scripts themselves forward the desired action (e.g. show next artefact) to Safari using an AppleScript-to-JavaScript bridge provided by AppleScript. In the first versions, the Wiimote's keystrokes where mapped to keystrokes of the Mac Pro's keyboard and the application in Safari was responsive to those. Soon, it turned out that this was insufficient since the keystrokes interfered with keyboard shortcuts of the operating system or Safari.

CONCLUSION

Finding a suitable hardware setup for *Preed* turned out to be more difficult than anything else. *Intensity Pro's* documentation is incomplete and the product is not very common. Although Blackmagic's support was very responsive, it took several weeks until it could be determined that the first card was defective. Also, *Intensity Pro* is very sensitive regarding the video input. When the card is fed with a signal coming from a Mini-DisplayPort-DVI adapter via a DVI-HDMI adapter and an HDMI cable, the screen remains black. When the two adapters are replaced by one Mini-DisplayPort-HDMI adapter, the signal is accepted. Still, the signal has to match the expectations of the card exactly, anything else will be ignored. However, this setup proved to be stable enough to meet the requirements of a experimental setup.

5.7 Exploring Preed with users

The user interface of the *Preed* prototype is shown in Figure 21 and Figure 22. The prototype was tested in a workshop with 12 students and a professor (Figure 23). At this point, the hardware setup was not yet available. For this reason the presentation view was opened in a browser window on the presenter's laptop and the Wiimote was used to switch between this browser window and the *Keynote* presentation. The students had to prepare examples of the use of typography and should show these artefacts using *Preed*. Consequently, all of the students had their laptops with them and turned on at the beginning of the workshop. They posted their artefacts, mainly PDFs and images, in the application. When it was their turn, they pushed their artefact to the presentation screen.

Figure 21: User interface of *Preed* (reconstructed)

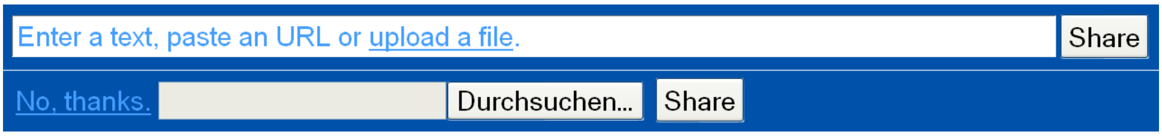

Figure 22: The two appearances of the publishing form. The upper row is the default appearance, a click on "upload a file" hides this row and shows the lower row instead. Another click on "No, thanks." brings back the upper row.

Figure 23: First round of testing

The results were very interesting. Several adjustments were made in the application, not counting software bugs that were also discovered. Maybe most important, the list of artefacts was completely overhauled. In the tested version, the artefacts were listed in the order of their publication. However, during the workshop it turned out that finding a previously presented item is a frequent task. Since the artefacts were not always presented in the order of their publication but more with regard to their content, this order by publication date was of little value. For this reason, the list was split in two lists that are arranged above each other. The first one is a global list of all artefacts that were pushed to the presentation screen, in the order of their appearance. This list is the equivalent of the browsing history in a web browser, but consists of artefacts instead of URLs. Interestingly, this is still a temporal criterion for sorting, but still very different from the news feeds of Twitter and Facebook, which were the starting point for this kind of content structure.

To that, a second list was added. This list shows the artefacts that were published by a certain user. By default, the artefacts of the current user are shown. Using a dropdown field the user can switch between the users. Artefacts that were published by the current user or were already made visible on the presentation screen can be opened by clicking on one of them. Items of other users which have not been shown yet are greyed out. In this second list, the filename of an uploaded file or the beginning of a URL or text is displayed instead of the name of the posting user. Since a certain artefact can occur multiple times in the two lists, not only the selected link is marked, but also all links to the same artefact regardless in which list they occur. Using these two lists, a user can find a previously visible artefact quickly for a second glance or for re-pushing it to the presentation screen. For image artefacts, retrieving is also made easier by replacing the icon for the image type with a preview of the image. Figure 24 gives an impression of these changes.

Figure 24: User interface of *Preed* for the second round of testing. Since the current user is not "peterpur", some items are greyed out in the lower list.

Against all expectations, the list of all artefacts on the presentation screen was hardly ever used. The presenter as well as the participants agreed spontaneously on who should push which artefact to the presentation screen. Since screen space is precious, it was decided to remove the list of artefacts in presentation mode by default and assign a button on the Wiimote to showing it as an overlay.

As another consequence of the test, a delete icon was added making it possible for users to delete their own artefacts as long as they were not shown on the presentation screen.

When the application was used on the high definition screen for the first time, it became obvious that images have to be sized to full screen automatically. The participants want to comment on the picture and therefore, it has to be displayed at an appropriate size regardless of its original resolution.

After making these changes and fixing some software bugs, a second round of testing was conducted. It took place in the same room but with a different group of people. The researchers of the Human Computer Interaction group present their current fields of research in a weekly lunchtime meeting and invite colleagues for discussions und presentations. Although people are urged to bring their own lunch, 6 out of 17 had their laptops on the table (Figure 25) during the lunchtime meeting *Preed* was tested in. A colleague had prepared a presentation about his research topic and started presenting. Since it is very common in this kind of meeting to ask questions and contribute to the presentation, a lively discussion was started soon. Especially the professor responsible for the presented project published content. When a topic popped up in the discussion, he tried to help the presenter by adding content to illustrate a point. For example, when the habits of the hip hop culture were discussed, several images showing musicians and fans were published. In this way, something like a moodboard of a discussion was created.

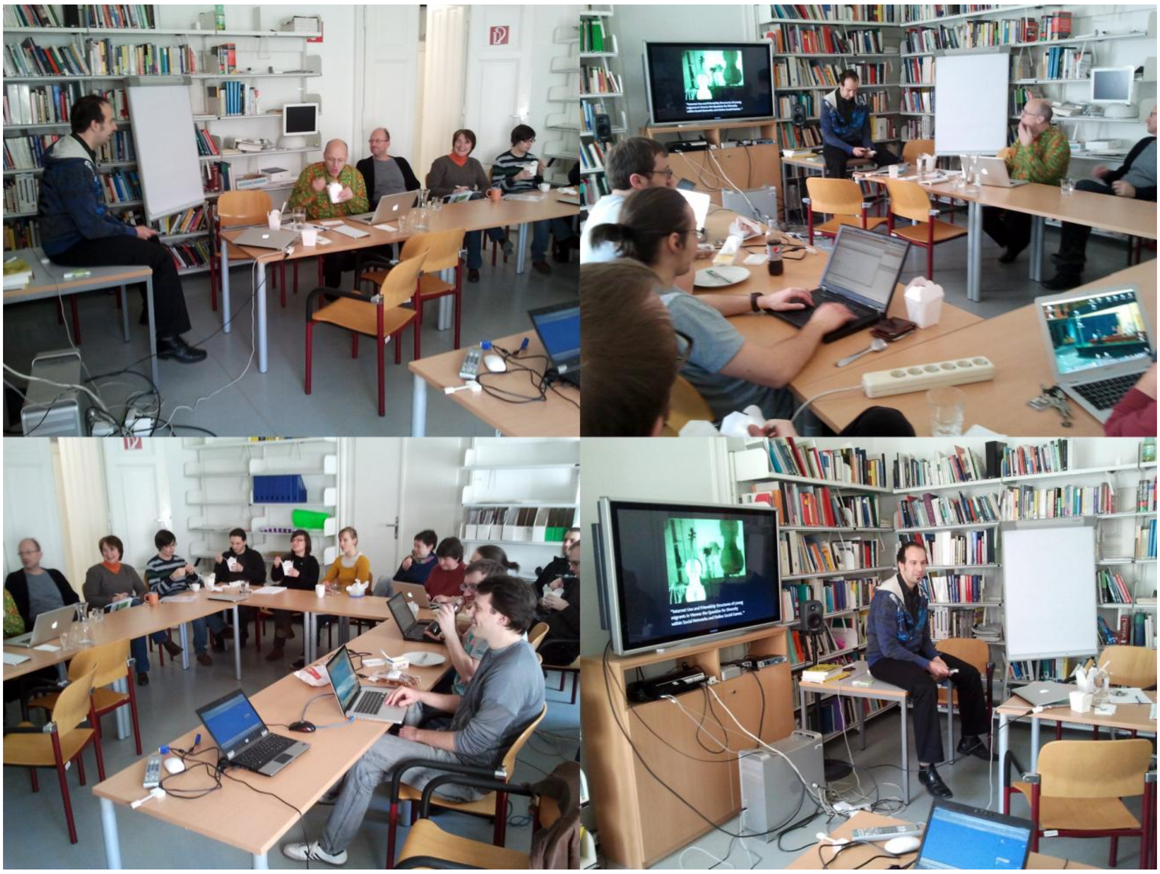

Figure 25: Second round of testing

Using *Preed* in a different context led to new lessons about how it might be used and which interactions were not yet clear. A big issue was the lacking feedback of the application to actions of the users. When a user posted an artefact in the tested version, the only feedback of the application that the upload had finished was the again empty publishing form. This led to several duplicated postings. Additionally, there was no feedback when pushing an artefact to the presentation screen. Since the presentation was visible on the presentation screen most of the time, the pushing user did not get any feedback that his action was actually carried out. Both issues were resolved by showing the artefact on the user's screen immediately after the upload had finished or the push action had been completed.

Unfortunately, it was discovered that some websites use JavaScript to direct the web browser running *Preed* away from the current URL. Since this cannot be prevented by the application itself, the home button on the Wiimote was enabled to reset the web browser, load the latest workspace and remove this item. This way, the presenter can restart the application by pressing one single button.

It was also noticed that the presenter had difficulties using the remote control. He was using a smartphone to control the presentation software. It is possible to gesture with one hand while holding a device in the other, but *Preed* required using another device to control the switching between his presentation and the participants' content. The participants had to tell the presenter to switch to their content; it never occurred that the presenter did not fulfil this wish during the test. Following the principle to give the group credit for the control of the screen, *Preed's* behaviour was changed: When a user pushes an artefact to the screen, it appears automatically there – without intervention of the presenter. However, the presenter still has the possibility to switch back to his presentation using the Wiimote and discussing inappropriate behaviour with the pushing user.

The hardware setup was used for the first time in this test round and proved to be functional and reliable.

5.8 Reflections on the development process

Designing *Preed* was a long and rocky path. The idea of supporting file sharing in a university workshop was developed into an elaborated collaboration studio and finally, an application to share and present digital artefacts was developed. To me, this broad range of different ideas in the same setting exemplifies a critical point in design. That there are an endless number approaches for a solution for a problem. This might sound like a platitude. In my opinion, it can often be observed in product development that the self-evident solution is not the best one. The first of *Preed's* predecessors was a kind of a graphical user interface for peer-to-peer file sharing with the peers being in the same room. While this served the purpose perfectly well, more research and exploration unveiled that for this specific context, the issue of presentation needs to be taken into account as well.

I would like to give another example: Shortly after the launch of Apple's iPad in early 2010, several manufacturers started offering all variants of protective cases. Many of these cases not only protect the screen, but also help erect the device to make typing on the virtual keyboard easier. I have hardly ever seen an iPad without such a case. A lot of iPad clones copied this idea of protective cases. When the iPad 2 was presented in early 2011, Apple had brought this kind of protection cover to the next step: The cover is magnetically attached to the device and can be folded in a way that it acts as a stand of the device. While it is pretty obvious that protecting the screen does not require a whole case, not a single of the other manufacturers (at least of my knowledge) had presented a similar idea. In my opinion, this is a perfect example of design innovation that takes the needs of the users seriously and is aware of the actual use of a product.

Returning to *Preed*, it can be said that the long exploration phase helped a lot in developing the tool. In the beginning, I was participating in the situation that I wanted to support with a tool. It was necessary to take a step out of this scenario, take a more general look at it and explore different ideas.

Sketching these ideas proved to be very useful. As soon as the first code lines have been written, it becomes very difficult to abandon an idea; even if it is obvious that it is not a good one. When the interviews with prospective users revealed that they are not able to bring up a situation of their daily lives where they would use such a tool, a working prototype had already been developed. It was very hard to accept these "sunken costs" and start all over again with another presentation paradigm (and another technology, in the end).

The user tests served a completely different purpose. While the sketching phase determined the direction to go, the results of the tests showed which shoes should be used. The changes resulting from the user tests were considerable, but the direction as such was not questioned again. In my opinion, Bill Buxton's view on the product development process, where design is not done in a self-contained phase but is thought of as an ongoing process with differing intensities in the process, is an appropriate one. Of Alan Cooper's *Goal-Directed Design* method, only elements were used. Personas, scenarios and ethnographic research helped to explore different options and ideas. However, the personas were not used to justify design decisions. I think the reason for this is that there were very few design decisions since the resulting product hardly has any features and workflows compared to common software applications. Additionally, it was possible to make design decisions based on user tests. This is seldom possible in a business setting, where software development is done in another department or by a different company. In this case, it is necessary to define the resulting application in advance. In my opinion, this is the reason why small start-ups, where designers and developers share the same room often come up with flexible and innovative products while big companies fail.

Concluding, a broad range of solutions was considered and one was finally chosen and prototypically implemented. It cannot be proved that the other solutions would have failed for sure. But the research and exploration phase allowed to thoroughly observe the usage context and to develop a solution that takes its requirements into account. The user tests were useful to refine the proposed solution, but did not question it as such.

6 Preed

Preed is a hardware and software setup that allows sharing and publishing of digital artefacts like images, websites, texts etc. and interaction with it. *Preed* is meant to support working with digital content in a creative context, i.e. a university tutorial or a brainstorming. *Preed* works transparently, i.e. in a way that whoever takes up the role of presenter does not notice a difference from common setups like a presenter connecting his/her laptop to a presentation screen.

While the last section described the process of design and development of *Preed*, this section will give an in-depth description of the final application, its features and how they are implemented. Following Google's principle to "put the users first" [39], the interface and interaction between *Preed* and its users is described. Subsequently, the implementation is presented and discussed.

Two terms will occur frequently in this chapter, presenter and user. The presenter is the person who connects his/her laptop to the *Preed* setup; the users are the ones who contribute content and publish it on the presentation screen. The content that is published will be referred to as "items" or "artefacts".

6.1 Interface and interaction

In order to work transparently, no software installation is required, neither on the presenter's nor the users' side. Given that the setup is already up and running, the presenter enters the room and connects his/her laptop with a provided hardware interface like in a traditional setting. In the reference setup of this work, a Mini-DisplayPort-adapter or a Mini-DVI-adapter respectively is provided and connected to the laptop. The presenter has to configure his operating system to output a signal. The reference setup only supports 1080p50, i.e. frames of 1920 by 1080 pixels at 50 hertz.

Starting *Preed* for the presenter only requires a single step: The presenter has to enter a name for the workspace. When the presenter powers up the presentation screen, it shows a blue screen with a single input field labelled with "URL of Collection" (Figure 26). This label indicates that only URL-compatible characters can be used. Using a mouse and a keyboard in the room, the presenter has to enter a name. When "OK" is clicked, the input field disappears and a text in quotation marks shows up (Figure 27): "To share content here, go to http://feed.bazalka.at/patterns", where "patterns" is the name of the collection the presenter has just chosen and "http://feed.bazalka.at/" is the domain where the server running the *Preed* application can be accessed. After the users have loaded the URL or written it down somewhere, the presenter presses "1" on the provided Wiimote (Figure 19) to show the signal of his/her laptop on the presentation screen like in a traditional setting.

Figure 26: Setting up a new workspace

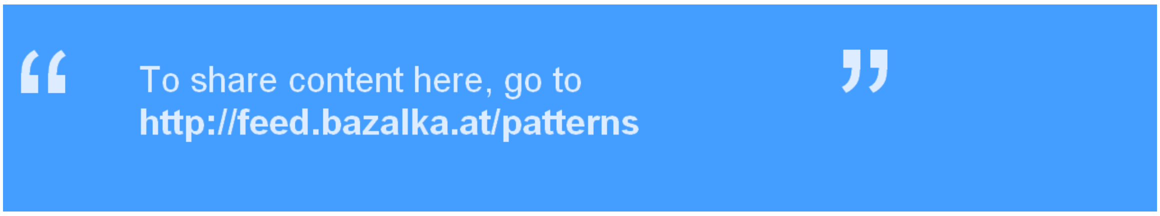

Figure 27: The first message of *Preed* on the presentation screen

The users only need a recent web browser to access the provided URL. The application was tested with Safari 5 and Firefox 3, which were the most common browsers at the university at the time of writing. When a user loads the URL in a browser, a single input field appears labelled with “Your name” (Figure 28). After the user has entered a (nick) name and has clicked “OK”, the application loads. On purpose, *Preed* does not use cookies, passwords or any other identifiers. When a user reloads the website, s/he will be prompted the name again. If a user with such a name already exists, the user will be logged into this account.

Figure 28: Logging into a *Preed* workspace

In Figure 29, the interface is shown. Four areas can be differentiated: The content area uses most of the screen’s space. At the bottom of the screen, the publishing form is located. At the right hand, two lists are arranged above each other. The upper list is labelled with “History”, the lower one “Giselle’s Content”; “Giselle” is the name of the current user. This is the list of a certain user’s content. These four areas will be described by means of a typical workflow.

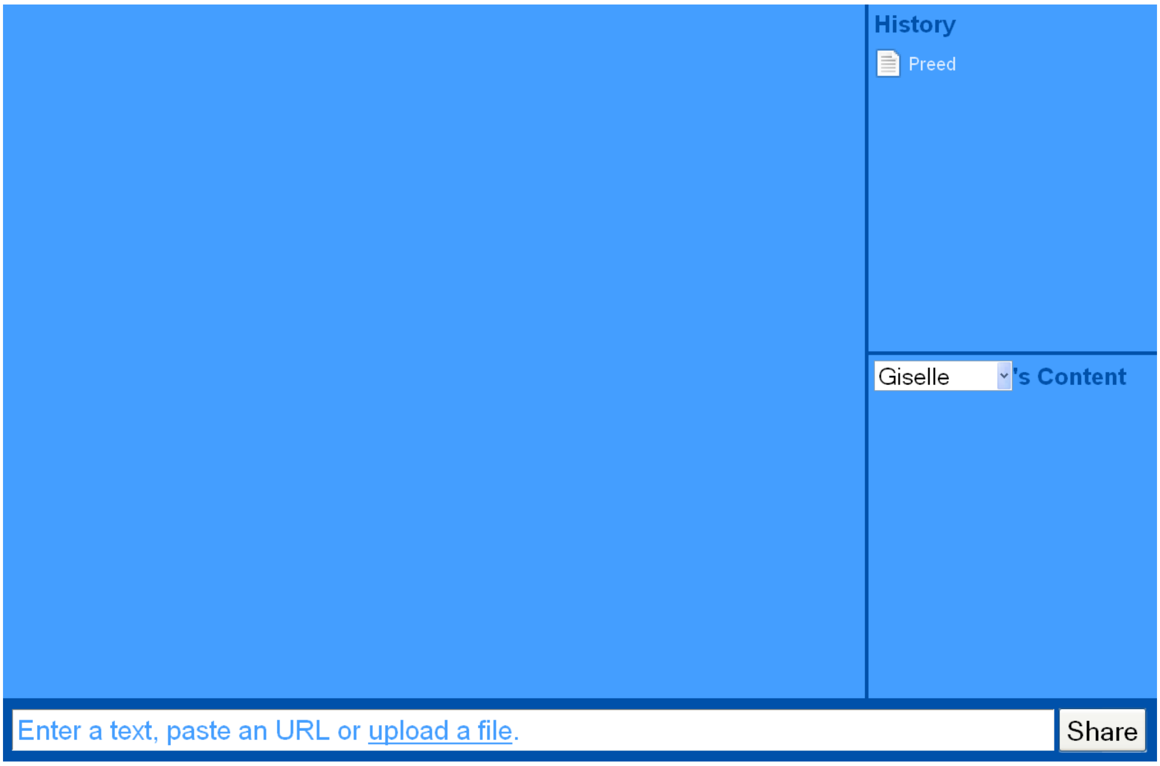

Figure 29: Accessing an empty *Preed* workspace

With *Preed*, the users are able to publish content on the presentation screen without the presenter taking any action. The first step to accomplish this is to publish the artefact using the publishing form. It is labelled with "Enter a text, paste an URL or upload a file", where "upload a file" is underlined and linked to a file upload form (Figure 30). When the user puts the cursor in the input field, the text disappears. So, if a user wants to show a website on the presentation screen to ask a question or support the discussion, the user simply pastes the URL of the website in the publishing form. After "Share" is clicked, the artefact is uploaded. It appears immediately in the content area and in the list of the user's artefacts (Figure 32). The artefact is described by the first letters of the shared URL and the favicon of the URL. If a user uploads an image (JPEG, PNG, GIF), a small preview of this image is shown instead of the favicon. The filename is used as the adjoining text. This is of course also true for any other file uploads. If the user enters a text, the first words of the text and a text icon will be shown. In the history list, the name of the contributing user is shown next to the icon, preview or favicon. Some of these variants are shown in Figure 31.

Figure 30: File publishing form

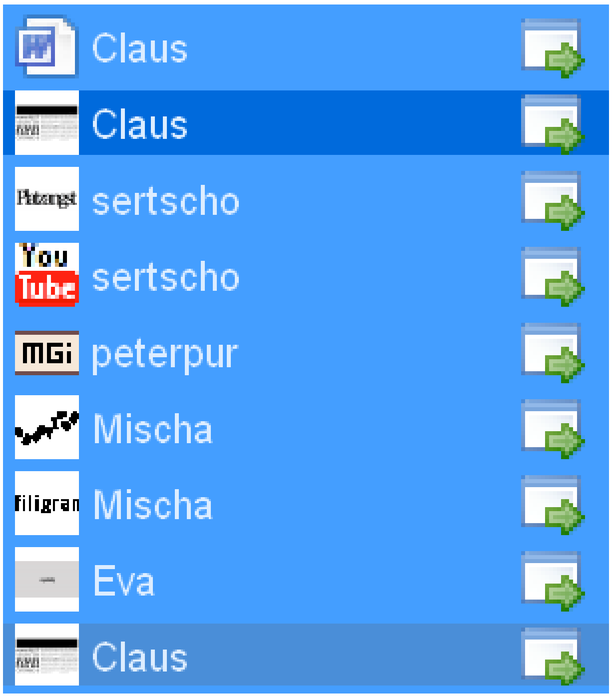

Figure 31: This clipping shows the history of a workspace. Every item can be pushed to the presentation screen by any user. The second most top item is displayed in the content area (not shown). The last list item in the list refers to the same content item. Every item has a preview item related to the file format or content type.

Figure 32: The user Giselle entered "http://tuwien.ac.at" in the publishing form and hit "Share". Thus, the website is shown in the content area and a new item is created in the user's list. It is shaded dark blue because it is currently shown in the content area.

When the user hovers the item with the mouse, the item is shaded dark blue. At this point, the user has three options to interact with the content. The user can click on it to display it in the content area. Alternatively, the user can click on the red delete icon to remove this item from the workspace permanently. This is only possible for items that have not been shown on the presentation screen yet. Finally, the user can click on the icon showing a screen, i.e. pushing this icon to the presentation screen. When this icon is clicked, *Preed* hides the signal of the presenter's laptop and displays the pushed artefact full screen without any surrounding interface. The presenter can then interact with the artefact using the Wiimote; this will be described in detail later in this section.

After being pushed, the artefact also appears in the history list of all users with its icon or preview and the name of the contributing user (Figure 31). New items are added on top of all other items. Any item in this list can be shown in the content area by clicking on it or can be pushed to the presentation screen. It is important to note that this is possible with all items in this list, not only with the items of the current user. When an item is pushed to the presentation screen another time, it is again added to the list unless it is already on top of it.

Figure 33: The user Giselle has clicked the push icon next to her single item. Therefore, the delete icon is removed and the item is added to the history. The dark blue shading of the item in the list of her items and the light blue shading of the item in the history indicate that these two list items refer to the same content item.

The second list, a list of a certain user's content is a support for finding a certain item again. The dropdown contains a list of all users of this workspace (Figure 34, left). Choosing a user from the dropdown causes the application to load a list of this user's items.

The items are displayed differently depending if they have already been shown on the presentation screen. Already presented items are displayed like the items of the current user, i.e. in a light blue colour. These items can be pushed to the presentation screen again by simply clicking the corresponding icon. The other items, i.e. the ones that have not yet been presented by the contributing user, are listed but greyed out. Unlike with the already presented items, it is not possible to click on these and inspect them in the content area. (Figure 34, right)

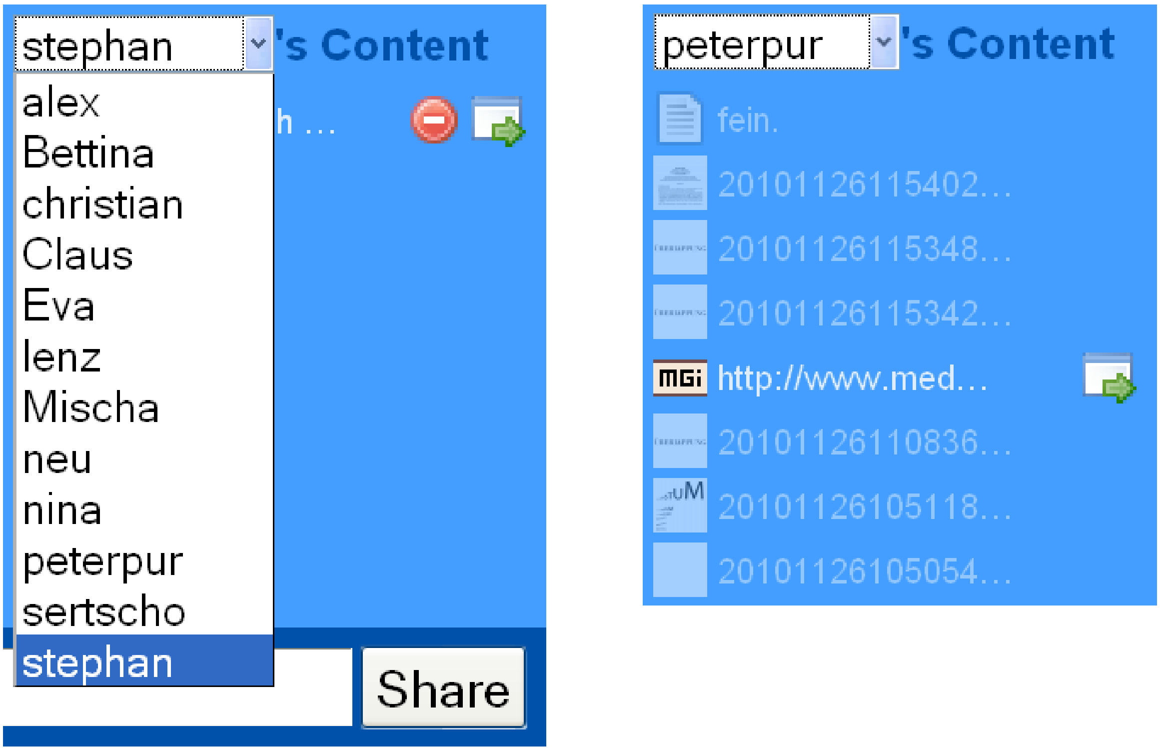

Figure 34: Selecting a user from the list of all users (left). Appearance of items of other users. Unpublished items are grayed out (right)

While the user has to possibility to interact with the artefact in the web browser, i.e. scrolling a website or downloading a picture, the presenter uses a Wiimote to interact with the content. *Preed* provides interaction possibilities for a number of formats which will be presented in the next paragraphs.

Uploaded pictures or URLs to pictures in the formats JPEG, GIF, and PNG are automatically sized to full screen. The presenter can use the plus and minus buttons on the Wiimote to zoom in or out of the picture. When the picture is bigger than the presentation screen, scrollbars are shown and the directional pad on the Wiimote can be used to show the desired clipping. These interactions are also valid for PDF documents. The pages of PDF documents are displayed as pictures and are shown page by page above each other.

The users are also able to publish a short text. The input field automatically adjusts its height. Text is displayed in quotation marks on the presentation screen. The plus and minus buttons on the Wiimote control the size of the text. Unlike the pictures, the text field is not resized but only the font size is changed. Thereby, horizontal scrolling is never necessary.

Preed also supports displaying other documents without the help of additional software. File formats that are supported by Google Docs Viewer are displayed inline. Among these formats are the most common Microsoft Office file types (DOC/DOCX, XLS/XLSX, and PPT/PPTX) and several image formats like TIF, PSD and SVG. Unfortunately, Google does not offer an API for the viewer. The Wiimote can be used to scroll with the directional pad, but resizing a document is only possible with the mouse of the server running *Preed*.

Files in other formats are displayed as links (Figure 35). The presenter can open *Preed* as well in a web browser, download the file and open it in a suitable application. Since the presenter is always able to make the signal of their laptop visible again, this is an option for files that are not supported by *Preed* natively.

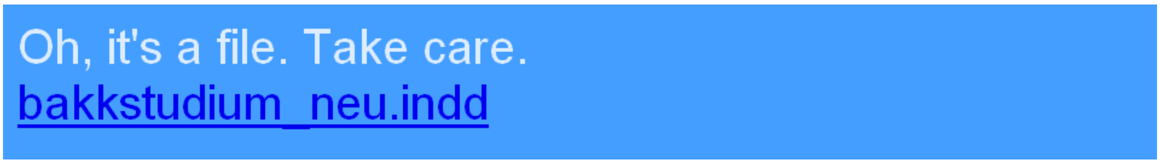

Figure 35: Displaying files in *Preed*.

Websites are generally displayed in an iFrame (Figure 33). It is possible to scroll the website using the Wiimote's directional pad and interact with the website using the mouse and the keyboard of the server running *Preed*.

URLs to YouTube are not displayed within an iFrame. YouTube provides application developers with a custom player based on the Flash Plug-in. Thus, *Preed* includes a small application from YouTube to display videos full screen instead of showing the full website with a lot of distracting content (Figure 36). The video itself can be played and paused using the A button on the Wiimote.

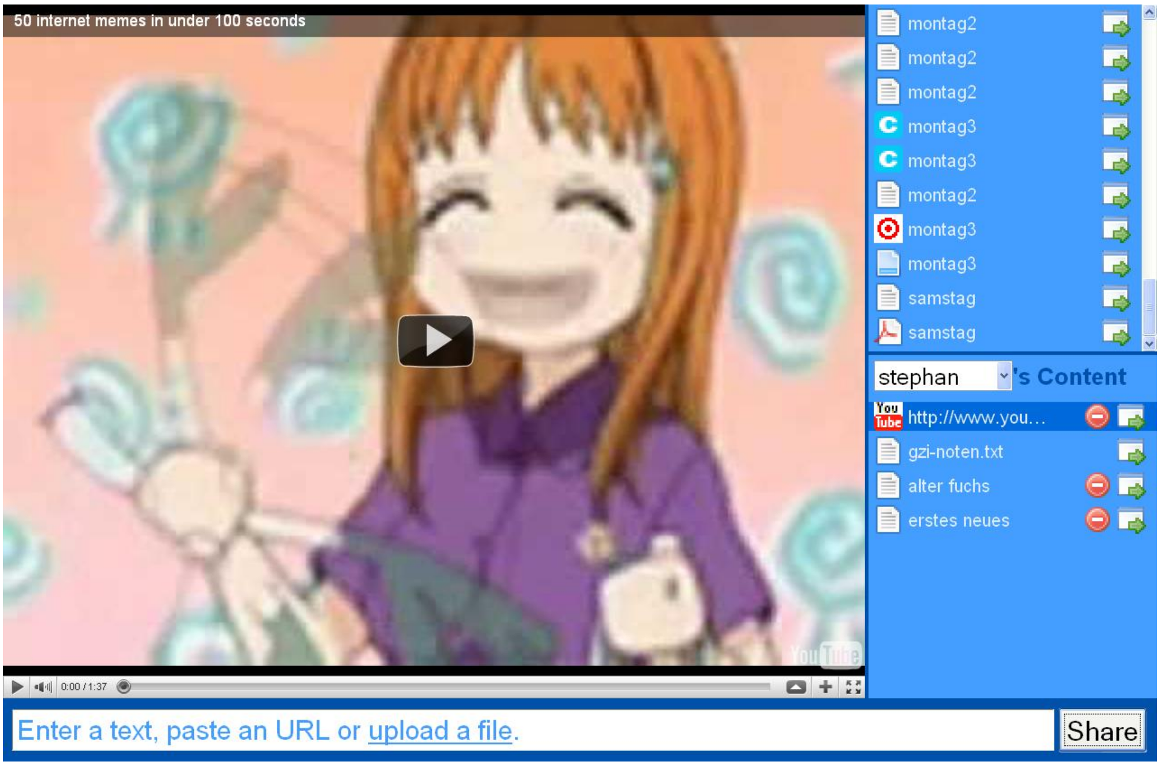

Figure 36: Displaying a YouTube video in the YouTube player.

To sum up, using *Preed,* users can share and publish digital content on a presentation screen. The presenter can interact with these artefacts and can still bring his/her presentation to the front again, using the Wiimote. *Preed* supports a range of common content types natively.

6.2 Technical perspective

This section will provide a description of the *Preed* reference setup from a technical perspective. To ensure the user experience described above, several elements have to be considered:

- A web application to allow the publishing of digital content.
- A device providing the web application to the users and receiving the signal of the presenter's laptop.
- A Wiimote for the presenter to interact with the content.

These three parts have to interact with each other. The next paragraphs will describe how this is accomplished.

In the reference setup for *Preed, a* Mac Pro acts as a web server. An Apache web server runs on the Mac Pro and provides the clients, i.e. the users' web browsers, with the *Preed* web application. The Mac Pro is also equipped with a Blackmagic *Intensity Pro*. This PCIe card provides an HDMI input and QuickTime drivers to process the input of high definition video. The presenter connects his/her laptop to the HDMI input of the *Intensity Pro* with a Mini-DisplayPort adapter and duplicates or extends the laptop's screen to the Mini-DisplayPort.

The Mac Pro's DVI output is connected to the presentation screen. Using a *Quartz Composer* patch running on the Mac Pro, the signal from the presenter's laptop is displayed on the presentation screen in a full screen window. Another full screen window is available. In this Safari window, the *Preed* web application is loaded in presentation mode.

As described in the previous section, a Wiimote is used to control which of these two windows is in front. It is possible to connect the Wiimote to the Mac Pro via Bluetooth and map its buttons to AppleScripts with a software tool named *OSCulator*. Pressing the buttons 1 or 2 on the Wiimote calls a corresponding AppleScript which tells either the *Quartz Composer* or the Safari window to come to the front.

The software part of *Preed* is implemented as an AJAX web application. For the part that is displayed in the user's web browser, HTML, CSS, and JavaScript are used. This frontend communicates with a PHP- and MySQL-driven backend via URL request and JSON-encoded messages. For display and communication utilities, the free JavaScript framework *MooTools* [40] is used.

The communication between the server and its clients is accomplished with JSON-encoded messages. A PHP script accepts a timestamp as input and returns an appropriate message, which contains six different kinds of information:

- The list of items which have been pushed to the presentation screen since the transmitted timestamp. These items are added to the list labelled "History".
- The list of items which have been published since the transmitted timestamp by the user which is selected in the dropdown menu. These items are added to the corresponding list.
- A list of all users logged into the current workspace since the transmitted timestamp. This list is displayed in the dropdown to show a certain user's content. New users are added to the dropdown.
- The list of items which have been pushed to the presentation screen since the transmitted timestamp. In presentation mode, this item is immediately shown in the content area. In standard mode, a push-icon is added and the delete-icon is removed if the user that is selected in the dropdown is the current user.
- The list of items which have been deleted since the transmitted timestamp. These items are immediately removed from both lists.
- Finally, a timestamp generated the moment the server sent its response to the client. This timestamp is saved and retransmitted with the next poll. The server's response only contains items and information that have been generated after this timestamp. Thus, new information can be transmitted incrementally instead of reloading all of the data the moment a new item is published.

The web browser polls this PHP script once a second in presentation mode and every five seconds in standard mode.

For every item, several attributes are transmitted, including

- the name of the contributing user,
- the timestamp of its publication,
- the content type,
- the icon to be used,
- the text that is displayed next to the icon,
- HTML code to display the content of the item.

This data is used to add items to the two lists as described above. When an item in the list is clicked on, the HTML code of the item is retrieved from the global item array and inserted into the content area. Some content types, i.e. YouTube videos or images, require some JavaScript code to be executed after the code has been inserted. This code initializes additional event listeners and save variables to the global item array.

When the user enters information into the publishing form and hits "Share", an asynchronous POST request is sent to the server. The PHP script receiving the request executes some basic security checks, writes the data into the database and sends an answer to the client. This causes the client to poll for the item script instantly to receive the newly shared item. Depending on the item's content type, the PHP script also executes some specific actions:

- Files that have been transmitted are generally renamed for security reasons and saved in a dedicated folder.
- If the file is an image format, a small preview is generated and saved.
- When a PDF file is uploaded, a short Perl script splits the file up into multiple single-page PDF files. The img-tag is used to display the pages of the PDF file one after the other. This tag has an undocumented feature in Safari under Mac OS X which allows displaying PDF files. Safari does not distinguish these PDFs which are displayed as images from ordinary images.

It is important to note that all of the other content specific processing like loading a custom player is done in the item script, not in the upload script. Although this causes more server load, it is a more flexible approach for an experimental setup. By doing the processing every time the item script is polled, an improved script has an immediate effect.

All other actions like pushing or deleting an item are executed by sending an asynchronous GET request to the server and polling the item script after a successful response has been received. In case of a push action, the PHP script sends a short AppleScript (via PHP's exec() and OS X's osascript) command to the operating system to bring the Safari running the *Preed* web application to the front.

Controlling the content inside of the *Preed* web application with the Wiimote is accomplished via *OSCulator*. This application maps every button on the Wiimote to a short

AppleScript. AppleScript provides an AppleScript-JavaScript bridge, meaning that a JavaScript function can be called from within an AppleScript. This means that there exists an AppleScript for every button on the Wiimote that calls an according JavaScript function inside the *Preed* web application.

In this section, the *Preed* software and hardware setup was presented. The software part consists of an AJAX-based web application using PHP and MySQL as a backend. This web application is responsible for presenting and collecting digital artefacts. The users access the web application with the web browser on their devices. They are able to upload new content, retrieve content of other users and publish content on the presentation screen. The server, a Mac Pro, which is connected to the presentation screen where the application runs, serves as a relay for the signal of the presenter's laptop. The presenter connects his/her laptop to the Mac Pro which is equipped with a particular extension card to process high definition video. Using a Wiimote, the presenter is able to interact with the users' content and to switch between his/her signal and the users' content. *Preed* proved to be an innovative and helpful approach for sharing and presenting digital artefacts in several rounds of testing and evaluation.

7 Future work

This section explores the possibilities for further work with *Preed*. Technological improvements will be presented along with ideas for future use.

Regarding the *Preed* web application, an anecdote explains the value of such a prototype. When Alan Cooper presented Ruby, a programming language that became part of Visual Basic, to Microsoft, the managers there were impressed and bought it. Cooper says that "the first thing I did was to throw the Ruby-the-prototype away and start over from scratch with nothing but the wisdom and the experience." [24] Although the responsible manager was furious about this action, ironically the final software was released in time whereas the accompanying Windows version was delayed a whole year.

The reference setup of *Preed* is highly experimental and its code has been reworked several times in order to get the user experience right. So, for use in a productive setting, it would most probably be necessary to start again – with the wisdom and experience gained during the process, of course.

A serious deficiency when it comes to using *Preed* in a productive setting is the fact that for several reasons it can only handle one format: Full HD at 50 Hz. Unfortunately, only one single HD format can be selected from a list of formats in the *Intensity Pro's* drivers. Nothing but this single format will be accepted. Thus, *Preed* only accepts full HD at 50 Hz although *Intensity Pro* would be able to capture other HD formats as well. Maybe this could be solved with a more elaborate *Quartz Composer* patch combining several (virtual) inputs.

Preed also lacks support for non-HD formats. The *Intensity Pro* only has an HDMI input, but many devices only support VGA. A video upscaler like *Spatz VGAHDMI* [41] with a VGA input which could upscale the video source and output HD via HDMI in a format acceptable to the *Intensity Pro* would solve this problem. Alternatively, an entirely different device with the ability to capture a wider range of formats and with VGA, DVI and HDMI interfaces might replace the *Intensity Pro*, as long as it is still possible to show the captured video in real time in a full screen window.

There is even more room for improvement in fields of technology. Because of its size, a Mac Pro is very unwieldy and difficult to hide in a cupboard. This is certainly possible with a Mac Mini, but this computer lacks a USB 3.0 interface, which is necessary for *Intensity Pro's* external brother, the *Intensity shuttle*.

In the reference setup, wireless internet was available for the users. If it is known that the meeting room lacks an internet connection, the Mac Pro or Mac Mini might act as a web server running the application itself as well. It could provide a wireless LAN connection to the participants' devices for connecting to *Preed's* user interface. However, this setup assumes that all of the content is already on the participants' devices or that it will be created "on the fly" since research in the World Wide Web is not possible.

Setting up *Preed* is cumbersome at the moment. The Mac Pro has to be powered on and a user has to log in. The Wiimote has to be connected via Bluetooth. Safari, OSCulator and Quartz Composer have to be started and configured. Additionally, choosing the name for the workspace with a different mouse and keyboard is a strange experience. The name of the workspace could also be defined automatically or from the presenter's laptop.

Regarding the software part of *Preed*, several desirable enhancements are obvious. Unfortunately, it was not possible to implement two desired core functions: Drag & drop and copy & paste. At the time of writing, drag & drop operations as defined in HTML5 were not commonly included not even in the latest web browsers. To provide a similar user experience in all web browsers, dragging files onto the publishing form is not supported even in browsers where this might be possible. Supporting copy & paste that goes beyond the transfer of text is not at all possible in current browsers. It is not possible to copy a part of an image from Photoshop and paste it into a web application using the HTML standard. However, this is a serious drawback of every web application which wants to handle data other than text.

Relying on Google Docs Viewer for the presentation of a lot of file formats can be considered a fill-in. Google does not provide an API for this tool, thus functions like zooming cannot be used with the Wiimote but only with a mouse. Scribd [42], a website for sharing documents, started converting a broad range of common file formats to HTML5. Unfortunately, Scribd only provides a Flash based viewer at the time of writing, but an HTML5 viewer will soon be available. Documents which cannot be displayed in the browser without dedicated software could be uploaded to a Scribd account automatically. The document is then rendered in HTML5 by Scribd, displayed in *Preed* using the Scribd viewer which could be controlled via the API with the Wiimote.

Concerning the use of *Preed*, two scenarios were brought up by colleagues who were part of the testing. They could not be more diverse. The first scenario is related to the YouTube integration. A colleague found this very impressive; he remembered parties where young people were discussing which YouTube music video should be played. They all gathered around the computer and looked for the favourite clips. Extending *Preed* by a mobile interface, it would have been easy to conduct the search individually and post the videos into a kind of queue. – The second scenario takes place at university and was discovered in the second round of testing. One colleague supported the presentation by looking for pictures, websites and videos that were related to the ongoing discussion. Instead of using *Preed* for direct involvement of students, a colleague suggested to use it as a support for the lecturer in a large course. When students ask questions or start a discussion, tutors could look up and publish information to support the lecturer in answering the question. These are only two of several scenarios that might be supported by *Preed* in the current state.

When reflecting about the original scenarios, it becomes obvious that tasks like brainstorming or event planning are only supported by *Preed* in a very narrow field: Presenting and sharing digital artefacts. However, these scenarios need more support, especially in the field of collaborating on digital content. There are tools for editing text [35], diagrams [36] or mind maps [43] collaboratively, but every tool needs different user accounts and implements different interaction principles. There is certainly a need for a tool that broadens the idea of sharing digital artefacts to collaborate on them.

Concluding, it was shown that the work at hand proved to be a helpful tool, but can only be considered the starting point for further exploration of how to support co-located meetings that have to deal with digital artefacts.

8 Conclusion

The thesis at hand describes design and development of a software and hardware setup that supports sharing and presenting digital artefacts in meetings. The impetus for *Preed* came from a university workshop I took part in some years ago. In this very collaborative course, we were struggling hard to share images, documents and websites, both with the whole group of fifteen students and our lecturer as well as with smaller subgroups. USB flash drives, Skype, mail and switching between the devices connected to the presentation screen are cumbersome ways to share digital artefacts. I wanted to find a way to support this use case with a solution that keeps the barrier of entry as low as possible.

At the beginning of this work, related work of the field of computer supported cooperative work as well as commercially available software are presented and discussed. There are very few products to support meetings with up to fifteen participants. This is very surprising, given the fact that such meetings are an everyday routine in the context of business and university. In both places, digital artefacts like images, websites, texts and documents play an important role. Commonly, meeting participants bring their devices, be it a laptop, a tablet or a smartphone, with them but they are most often used separately, i.e. everyone uses his/her private device. While many meetings rooms are equipped with basic hardware like a video projector or a presentation screen, this facility is mostly used like some years ago. A single user connects his/her device to the presentation screen and shows some kind of digital content. The presented research projects included *CoLab* and *i-Land,* which try to augment meeting rooms with information technology.

Next, the methods of designing such an application are discussed. The design and development of *Preed* has been influenced by two authors, Bill Buxton and Alan Cooper. The latter is the originator of the *Goal-Directed Design* method. This method takes archetypical users called personas as a starting point. These are based on ethnographic research and help making design decisions because the designer has a clear view of the prospective users and their needs. Personas and scenarios were used in the design and development of *Preed*. On the other hand, Bill Buxton outlines his views of the product development process, which is not separated into self contained phases like "design" and "implementation". Instead, he suggests that these activities are carried out in every step of the process, but that they are of different importance. For designing, he favours the intensive use of sketches. These are quick, disposable, and foster feedback. By their nature, they support the exploration of a wide range of ideas in a very short time. Sketching proved to be an important exploration tool in the design and development of *Preed*.

In the course of this work, the *Preed* application is only the final result. Several intermediate steps were taken to design an application that supports the scenario best. The first sketch was a kind of a graphical user interface for peer-to-peer file sharing. The basic information structure was a room plan of the meeting room. The users occupy the virtual seat that corresponds to the real one. They are able to drag and drop files from their device to the other seats or a shared area and make them accessible to their colleagues this way. Since this approach seemed too inflexible, a different approach was developed. After the development of personas and scenarios, *Virtual Workspace* was designed. *Virtual Workspace* is a persistent, infinite, collaborative canvas. It provides interfaces to work on different kinds of digital content, like browsing through photo albums, annotating documents and collecting research material. When this approach was discussed with prospective users, an astonishing discovery was made. They could not think of a situation in their daily work routine which could be supported by *Virtual Workspace*. Presumably, the arrangement of artefacts on an infinite canvas was too different from the users' common mental models of files and folders.

Finally, *Preed* was designed and developed. The reference setup for *Preed* consists of two parts, the hardware and the software. The latter provides a web based interface to the participants of the meeting. Using this interface, the participants are able to share and present digital artefacts like images, websites, notes or documents with the other participants. Without any action of the lecturer, a user can present one of his/her artefacts on the presentation screen, overlaying the lecturer's presentation. The users are also able to see artefacts of other users and a history of the artefacts that were presented previously. The hardware part is a Mac Pro equipped with a PCIe card which provides an HDMI input. The lecturer connects his/her device to this HDMI input. The Mac Pro is connected to the presentation screen and alternatively shows the signal of the lecturer's device or content from the participants. The lecturer can switch between these two sources with a Wiimote. This remote control is also used to interact with the content contributed by the users. Images can be resized, videos played and paused, websites scrolled, etc. To keep the barrier of entry very low, *Preed* does not require any software installation, neither on the participants' nor the lecturer's device. The application was prototypically implemented and further refined after two interesting tests with users.

However, *Preed* focuses on presenting digital content, but this is only one aspect of supporting co-located meetings. It was surprising to learn how few tools exist for such a common situation. Apart from presenting, sharing digital artefacts and collaborating on them are fields that certainly need more research. With the results of this research, the work practice of many people could be supported and advanced. Information technology has proved to support social contexts like communicating over a large distance in the blink of an eye, but it still has to show that this could also be done in the context of meetings. I am sure that the interesting part of this research still lies in front of us.

References

[1] H. ISHII AND N. MIYAKE, "Toward an open shared workspace: computer and video fusion approach of TeamWorkStation," *Communications of the ACM*, vol. 34, Dec. 1991, pp. 37-50.

[1] N.A. STREITZ, P. REXROTH, AND T. HOLMER, "Does "roomware" matter? \line Investigating the role of personal and public information devices and their combination in meeting room collaboration," *ECSCW'97 Proceedings of the fifth conference on European Conference on Computer-Supported Cooperative Work*, Norwell, MA, USA: Kluwer Academic Publishers, 1997, pp. 297-312.

[3] FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E.V., *"BSCW"* [Online]. Available: http://public.bscw.de/ [Accessed December 1, 2010].

[4] N.A. STREITZ, P. TANDLER, C. MÜLLER-TOMFELDE, AND S. KONOMI, "Roomware: Towards the Next Generation of Human-Computer: Interaction based on an Integrated Design of Real and Virtual Worlds," *Human-Computer Interaction in the New Millenium, Addison Wesley*, 2001, pp. 551–576.

[5] J. GRUDIN, "Computer-supported cooperative work: history and focus," *Computer*, vol. 27, 1994, pp. 19-26.

[6] R. JOHANSEN, *Groupware: computer support for business teams*, New York, London: Free Press, Collier Macmillan, 1988.

[7] WIKIMEDIA FOUNDATION, INC., *"Computer supported cooperative work"* [Online]. Available: http://en.wikipedia.org/wiki/Computer_supported_cooperative_work [Accessed December 30, 2010].

[8] P. COOK, C. ELLIS, M. GRAF, G. REIN, AND T. SMITH, "Project Nick: meetings augmentation and analysis," *ACM Transactions on Information Systems (TOIS)*, vol. 5, 1987, pp. 132–146.

[9] FOSTER, GREGG, *CoLab, Tools for Computer-Based Cooperation*, Berkeley, CA, USA: University of California at Berkeley, 1984.

[10] N.A. STREITZ, J. GEIßLER, J.M. HAAKE, AND J. HOL, "DOLPHIN: integrated meeting support across local and remote desktop environments and LiveBoards," *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 345–358.

[11] PALO ALTO RESEARCH CENTER INCORPORATED, *"CoLab"* [Online]. Available: http://www2.parc.com/istl/members/stefik/colab.htm [Accessed December 27, 2010].

[12] G. FOSTER AND M. STEFIK, "Cognoter: theory and practice of a colab-orative tool," *Proceedings of the 1986 ACM conference on Computer-supported cooperative work*, 1986, pp. 7–15.

[13] PALO ALTO RESEARCH CENTER INCORPORATED, *"Milestones"* [Online]. Available: http://www.parc.com/about/milestones.html [Accessed January 1, 2011].

[14] S. ELROD, R. BRUCE, R. GOLD, D. GOLDBERG, F. HALASZ, W. JANSSEN, D. LEE, K. MCCALL, E. PEDERSEN, AND K. PIER, "Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration," *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 599–607.

[15] J.C. TANG AND S. MINNEMAN, "VideoWhiteboard: video shadows to support remote collaboration," *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, 1991, pp. 315-322.

[16] NETOP SOLUTIONS A/S, *"Netop Vision6"* [Online]. Available: http://www.netop.com/de/produkte/education/netop-vision6.htm [Accessed January 5, 2011].

[17] NETVIEWER AUSTRIA GMBH, *"Netviewer"* [Online]. Available: http://www.netviewer.com/de/ [Accessed December 1, 2010].

[18] SMART TECHNOLOGIES ULC, *"SMART Board interactive whiteboards - SMART*

Technologies” [Online]. Available:
http://www.smarttech.com/us/Solutions/Education+Solutions/Products+for+education/Interactive+whiteboards+and+displays/SMART+Board+interactive+whiteboards [Accessed January 5, 2011].
[19] ADOBE SYSTEMS INC., *“Adobe Labs - JamJar”* [Online]. Available: http://labs.adobe.com/technologies/jamjar/ [Accessed January 5, 2011].
[20] W.W. ROYCE, “Managing the development of large software systems: concepts and techniques,” *Proceedings of the 9th international conference on Software Engineering*, Los Alamitos, CA, USA: IEEE Computer Society Press, 1987, pp. 328–338.
[21] F.P. BROOKS, *The Mythical Man-Month. Essays on Software Engineering*, Addison-Wesley Longman, 1995.
[22] B.W. BOEHM, “A spiral model of software development and enhancement,” *Human-computer interaction*, R.M. Baecker, J. Grudin, W.A.S. Buxton, and S. Greenberg, eds., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 281–292.
[23] *“Manifesto for Agile Software Development”* [Online]. Available: http://agilemanifesto.org/ [Accessed February 19, 2011].
[24] A. COOPER, *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*, Sams - Pearson Education, 2004.
[25] J. BLOMBERG, J. GIACOMI, P. SWENTON-WALL, AND A. MOSHER, “Ethnographic Field Methods and Their Relation to Design,” *Participatory design: principles and practices*, Routledge, 1993.
[26] B. BUXTON, *Sketching User Experiences: Getting the Design Right and the Right Design*, Morgan Kaufmann, 2007.
[27] D.A. SCHON, *The Reflective Practitioner: How Professionals Think In Action*, Basic Books, 1984.
[28] C. SNYDER, *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, Morgan Kaufmann, 2003.
[29] S. PUGH, *Total Design: Integrated Methods for Successful Product Engineering*, Prentice Hall, 1990.
[30] COOPER, *“Cooper Careers - Interaction Designer”* [Online]. Available: http://www.cooper.com/documents/cooper_careers_IxD.pdf [Accessed February 20, 2011].
[31] R.L. CAMPBELL, “Will the real scenario please stand up?,” *ACM SIGCHI Bulletin*, vol. 24, Apr. 1992.
[32] *“frykholm.se”* [Online]. Available: http://www.frykholm.se/rita.html [Accessed July 30, 2010].
[33] *“7.7 Drag and drop — HTML5”* [Online]. Available: http://dev.w3.org/html5/spec/dnd.html [Accessed February 6, 2011].
[34] ADOBE SYSTEMS INCORPORATED, *“Adobe AIR”* [Online]. Available: http://www.adobe.com/products/air/ [Accessed February 6, 2011].
[35] *“Piratepad Beta”* [Online]. Available: http://piratepad.net/front-page/ [Accessed February 26, 2011].
[36] *“Cacoo - Create diagrams online Real time collaboration”* [Online]. Available: http://cacoo.com/ [Accessed February 26, 2011].
[37] *“Twitterwall @ Mobile Monday Switzerland #chmomo | Flickr - Photo Sharing!”* [Online]. Available: http://www.flickr.com/photos/twen/5136789764/ [Accessed February 12, 2011].
[38] *“Blackmagic Design: Intensity”* [Online]. Available: http://www.blackmagic-design.com/products/intensity/ [Accessed September 26, 2010].
[39] *“Santa Monica - US jobs - Google”* [Online]. Available: http://www.google.com/intl/kn/jobs/uslocations/santa-monica/index.html [Accessed February 27, 2011].
[40] *“MooTools - a compact javascript framework”* [Online]. Available: http://mootools.net/ [Accessed February 27, 2011].

[41] *"SPATZ VIDEO"* [Online]. Available: http://www.spatz-tech.de/upconver.htm [Accessed February 13, 2011].
[42] *"Scribd"* [Online]. Available: http://www.scribd.com/ [Accessed February 27, 2011].
[43] *"Mind42.com - Collaborative mind mapping in your browser"* [Online]. Available: http://mind42.com/ [Accessed February 27, 2011].