

Collaborating with Tangible Windows

*Idea Generation and
Information Exchange
in Small Groups*

Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University



by
Mario Fraikin

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Klaus Wehrle

Registration date: Nov 09th, 2010
Submission date: May 09th, 2011

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, May 2011
Mario Fraikin

Contents

Abstract	xv
Überblick	xvii
Acknowledgments	xix
Conventions	xxi
1 Introduction	1
1.0.1 Research Goals	2
1.1 Tangible Windows	3
1.1.1 The Vision of Ubiquitous Computing	4
1.1.2 Nomadic Operations	6
1.2 Iterative User-Centered Design	7
1.3 Chapter Overview	9
2 Collaborative Work	11
2.1 The Definition of Collaboration	12
2.1.1 Informing	12

2.1.2	Coordinating	12
2.1.3	Collaborating	13
2.1.4	Cooperating	13
2.1.5	Summary	13
2.2	Groupware	14
2.3	Characteristics of Collaboration	15
2.4	Brainstorming	17
2.4.1	Problem Formulation	18
2.4.2	Participants	19
2.4.3	Organization	20
2.5	Limiting Factors for Brainstorming	21
2.5.1	Production Blocking	21
2.5.2	Evaluation Apprehension	22
2.5.3	Free Riding	22
2.5.4	Discussion	23
3	Related Work	25
3.1	Augmented Work Environments	26
3.1.1	i-Land	26
3.1.2	iRoom	29
3.1.3	The NiCE Discussion Room	30
3.2	Middleware	32
3.2.1	ARIS	33

3.2.2	IMPROMPTU	34
3.3	Meeting Software	36
3.3.1	TEAM STORM	37
3.3.2	ThinkTank	39
4	Task Definition	41
4.1	Survey: Collaborative Work in Business Meetings	42
4.1.1	Participants	42
4.1.2	Analysis	43
	Meeting Topics	44
	Meeting Structure	45
	Group Size	46
	Productivity Hindrances	48
	Improvements for Collaboration	50
	Tools	51
4.1.3	Results	54
4.2	Paper-Based Brainstorming Session	54
4.2.1	Participants	55
4.2.2	Set-Up	55
4.2.3	Observations	57
4.2.4	Results	59
4.3	Summary	61

5	Storyboards	63
5.1	Situations	64
5.1.1	Parallel Idea Contribution	64
5.1.2	Idea Access	65
5.1.3	Delegation of Work	68
5.2	Evaluation	70
5.2.1	Participants	70
5.2.2	Set-Up	70
5.3	Results	71
6	Software Prototype	73
6.1	Design Requirements	74
6.2	System Design	75
6.2.1	Input Modes: Sketching and Typing .	78
	Sketching	79
	Typing	80
	User Interface Improvements	81
6.2.2	File Handling	83
6.2.3	Correction and Error Recovery	83
6.2.4	Organization	85
6.2.5	Information Exchange	87
6.3	System Implementation	88
6.3.1	Nomadic Operations Framework	89

6.3.2	Class Model	90
6.3.3	Implementation Challenges	93
	Coordinate Systems	93
	Text Handling	94
7	Evaluation	97
7.1	Qualitative User-Study	97
7.1.1	Participants	99
7.1.2	Set-Up	99
7.1.3	Observations	102
	Idea Generation Phase	102
	Idea Presentation Phase	102
7.1.4	Retrospective Interviews	104
7.1.5	Results	106
7.2	Summary	110
8	Summary and Future Work	113
8.1	Summary and Contributions	114
8.2	Future Work	116
8.2.1	Software Improvements	116
	Idea Organization	116
	Additional System Features	117
	Additional User Studies	117

8.2.2	Additional Application Scenarios . . .	118
8.2.3	Nomadic Operations	119
8.2.4	Additional Form Factors for Tangible Windows	120
A	Survey Questions: Collaborative Work in Business Meetings	123
B	Questionnaire for the Sketch It! User Study	129
C	Storyboards	133
	Bibliography	141
	Index	145

List of Figures

1.1	Tangible Windows	4
1.2	DIA Cycle	8
2.1	Collaboration Cycle	15
3.1	i-Land: CommChairs	28
3.2	iRoom	29
3.3	The NiCE Discussion Room	32
3.4	ARIS	34
3.5	IMPROMPTU	35
3.6	TEAM STORM	38
3.7	ThinkTank	40
4.1	Diagram: Meeting Participation Rates	43
4.2	Diagram: Group Sizes	47
4.3	Diagram: Participant Satisfaction	49
4.4	Diagram: Meeting Tools	52
4.5	Brainstorming Session Set-Up	56

4.6	Brainstorming Result	60
5.1	Storyboard: Classical Brainstorming	66
5.2	Storyboard: Tangible Windows Brainstorming	67
5.3	Diagram: Storyboard Evaluation Results . . .	71
6.1	Splash Screen and Icons	76
6.2	Sketch It! Screenshots	77
6.3	Tools and Properties	79
6.4	Revised GUI elements	82
6.5	File Selection	84
6.6	Collection of Minimized Sketches	86
6.7	Tool Window	87
6.8	Class Diagram	91
6.9	On-Screen Keyboard & Text Elements	96
7.1	User Study Set-Up	100
7.2	Grouping of Sketches	103
7.3	Diagram: User Satisfaction with Sketch It! . .	108

List of Tables

7.1	Satisfaction & motivation of the participants	107
7.2	Evaluation of equipment & Nomadic Operations	109

Abstract

Interaction with modern operating systems is based on the window metaphor — applications embed their content in virtual rectangles that are spread over the computer screen. The Tangible Windows concept is an alternative approach, which takes the windows from the screen and brings them into the physical world. It utilizes several device classes such as tablet computers and wall displays to let users interact with data independent of a single device. We denote the operations used to exchange information between various Tangible Windows “Nomadic Operations”.

In this thesis, we apply the Tangible Windows concept to a collaborative scenario: brainstorming. In an iterative design process, we have developed the fully functional software “Sketch It!” that was implemented for Apple iOS and Mac OS X providing tools for note taking and idea sketching. We have conducted a survey investigating the frequency of today’s meetings in business companies and collected information about tools and equipment utilized in these situations. Furthermore, we analyzed advantages and drawbacks of group work. Observations of a paper-based brainstorming session led to additional insights into the interaction between meeting participants and their workflows.

Based on these findings, we designed a first prototype realized as storyboards to illustrate potential use cases of the software and evaluated their plausibility with several people. We then present the software prototype and a set of design requirements that shaped its development. The system implementation consists of the applications “Sketch It! for iPad” and “Sketch It! for Mac”, which are illustrated including applied improvements as well as our solutions for implementation challenges that occurred during development.

A qualitative user study of the Sketch It! system was arranged as a brainstorming session of collaborating employees from a business company. The study showed that Nomadic Operations are suitable support for collaborative tasks. The meeting participants indicated high interest to incorporate the Sketch It! system into their future brainstorming sessions.

Überblick

Die Benutzerinteraktion mit modernen Betriebssystemen basiert auf der Fenster-Metapher — Applikationen betten ihre Inhalte in virtuelle Rechtecke, die über den Computerbildschirm verteilt werden. Das Tangible Windows-Konzept stellt einen alternativen Ansatz dar, indem es die Fenster von dem Computerbildschirm in die physische Welt bringt. Durch verschiedene Geräteklassen wie Tablet-Computer oder Wand-Displays lässt es den Benutzer mit Daten interagieren, die unabhängig von einem einzelnen Gerät sind. Wir bezeichnen die Operationen zum Informationsaustausch zwischen Tangible Windows als "Nomadische Operationen".

In dieser Arbeit übertragen wir das Tangible Windows-Konzept auf ein Gruppenarbeit-Szenario: Brainstorming. In einem iterativen Entwicklungsprozess haben wir mit "Sketch It!" eine voll funktionstüchtige Software entwickelt, die für Apple iOS und Mac OS X implementiert wurde und Werkzeuge zur Erstellung von Notizen und Skizzen zur Verfügung stellt. Wir haben eine Umfrage durchgeführt, um die Häufigkeit aktueller Meetings in Geschäftsunternehmen zu untersuchen und Informationen über Werkzeuge und die Ausstattung für diese Situationen zu sammeln. Außerdem haben wir Vor- und Nachteile von Gruppenarbeit analysiert. Die Beobachtungen einer papierbasierten Brainstorming-Session führten zu zusätzlichen Einblicken in die Interaktion zwischen Meeting-Teilnehmern und ihren Arbeitsabläufen.

Basierend auf diesen Erkenntnissen haben wir eine erste Prototyp-Umsetzung erstellt, die in Form von Storyboards potentielle Anwendungen der Software illustriert. Deren Plausibilität haben wir mit mehreren Personen evaluiert. Anschließend präsentieren wir den Software-Prototypen und eine Menge von Design-Anforderungen, die den Entwicklungsprozess geprägt haben. Die System-Implementierung besteht aus den Applikationen "Sketch It! for iPad" und "Sketch It! for Mac", welche inklusive vorgenommener Verbesserungen sowie unseren Lösungen von Implementierungsherausforderungen erläutert werden.

Eine qualitative Benutzerstudie des Sketch It!-Systems wurde als Brainstorming-Session für gemeinschaftlich arbeitende Angestellte eines Geschäftsunternehmens durchgeführt. Die Studie hat gezeigt, dass Nomadische Operationen geeignete Unterstützung für gemeinsames Arbeiten darstellen. Die Meeting-Teilnehmer hatten großes Interesse, zukünftig das Sketch It!-System zum Brainstorming einzusetzen.

Acknowledgments

First of all, I want to thank Prof. Dr. Jan Borchers and all the people working at *The Media Computing Group* for their assistance. Everyone at the chair showed immediate cooperativeness and contributed valuable help to my project.

Special thanks to my advisor Jonathan Diehl, who supported me in a timely manner and consistently provided constructive feedback throughout my work.

Sören Busch assisted me with the conduct of the user study and took part in many insightful discussions. Thanks for your support!

Furthermore, Chatchavan Wacharamanotham helped me with the design of my surveys and user studies. I am very thankful for your help.

I also want to give thanks to Christine Murakami and Roland Hildebrandt for diligently reviewing my thesis and providing competent advice.

The participants in my surveys as well as the people who took part in the user studies and evaluations put much effort into them and spent precious time, which I highly appreciate.

I thank Jill, who encouraged me all along my studies. Finally, I am very grateful to my parents, who were always there for me.

Conventions

Throughout this thesis I use the following conventions:

The whole thesis is written in American English.

Important terms are written in *emphasized typeset* when they appear in the text for the first time.

There are two characters I refer to in this thesis: the designer and the user. In order to simplify dealing with these roles, I decided to identify the designer as male and the user as female.

Chapter 1

Introduction

*“I think I could, if I only knew how to begin.
For, you see, so many out-of-the-way things had
happened lately that Alice had begun to think that
very few things indeed were really impossible.”*

— Lewis Carroll

In companies with a certain amount of employees, meetings are an established part of the working schedule. Small groups of people exchange their thoughts, brainstorm about new project ideas and develop concepts for the implementation of their new products. But meetings are not limited to business organizations: students, pupils and private groups also come together to collaborate on a diversity of tasks.

Meetings are part of
the daily life

There are numerous ways to conduct a meeting, retain its outcomes and structure the collaborative work. A common approach to the exchange and collection of information in today’s meetings is to keep record of the participants’ contributions in text documents (either digital or physical) or write them onto a whiteboard. Often, one user needs to merge others’ ideas and chooses email as channel for providing the outcomes to all meeting participants. There are also more sophisticated solutions such as dedicated meeting environments with multi-touch tables (tabletops) or interactive wall screens providing computer-based support

for the collaboration.

Users of these systems are often limited to specific devices and environments, which influences the collaborative workflow. Computer support for collaborative work has to meet challenges with respect to natural interaction and information exchange between devices.

Tangible Windows is a flexible and natural interaction concept

In this thesis, I will investigate how meeting participants make use of a software solution based on a new interaction concept called *Tangible Windows (TW)*. The Tangible Windows concept is an approach to turn the well-known digital windows metaphor taken from modern Operating Systems' desktop into hardware. The devices constituting the windows are connected through a network infrastructure and the data generated by users can be freely exchanged between these devices. This natural interaction concept creates flexibility in the way people generate, combine and move their data: users are no longer restricted to a specific device when they create and interact with information. Additionally, they are provided with a simple way to share their content with other users through different kinds of equipment suitable for the current task. I will fully describe the concept in the following section.

The prototype will influence the collaborative workflow

My software prototype, called "Sketch It!", is a sketching software with additional support for text editing. It was implemented for [Apple iPads](http://www.apple.com/ipad/)¹ and [Mac OS X](http://www.apple.com/macosx/)² systems featuring parallel idea generation and information exchange. It was evaluated in a brainstorming session, which is an example for a task present in various collaborative meetings. During my work, I focused on the adaption of the Tangible Windows concept to a collocated scenario indicating that meeting participants collaborate face-to-face while being in the same environment.

1.0.1 Research Goals

In this thesis, I will give answers to the research question if the Tangible Windows concept is able to support the users'

¹<http://www.apple.com/ipad/>

²<http://www.apple.com/macosx/>

productivity in their collaboration. Furthermore, I will explore how users feel while using the system and if they perceive the technology as a benefit to their business.

The outcomes of collaborative work will depend on the kind of task the group deals with. It needs to be investigated how the TW concept is able to contribute tools that support the users with their specific requirements. My contribution will be an evaluation of the appropriateness of Tangible Windows for collaborative work. Additionally, I will provide a collection of guidelines how to design software components supporting collaboration based on TW.

My research contribution is a collection of design guidelines for collaborative TW systems

In the following sections, I will explain the concept behind Tangible Windows and the development approach *Iterative User-Centered Design* since this approach was adopted in my work and exerted a powerful influence on it.

1.1 Tangible Windows

Modern Operating Systems for desktop computers are based on the window metaphor: application content is embedded into virtual rectangles that are spread over the computer screen.

Tangible Windows is an alternative interaction concept. The virtual windows from the desktop are turned into physical devices as illustrated in Figure 1.1. The users can hold them in their hands, take the windows with them, and integrate the devices into the environment. Tangible Windows have the advantage of providing natural affordances to their users: Norman [2002] describes affordances as strong clues to the operations a user can do with objects. Natural behavior such as holding, moving and showing of Tangible Windows gets triggered by their physicality.

Tangible Windows are virtual windows turned into physical devices

The approach of focusing on natural and flexible usage of a device collection suitable for a diversity of tasks is based on the vision of *Ubiquitous Computing* that I explain in the following section.



Figure 1.1: The Tangible Windows concept

1.1.1 The Vision of Ubiquitous Computing

Ubiquitous Computing is the disappearance of computer technology into the environment

In his article *The computer of the 21st century* [1999], Mark Weiser predicts a change of the interaction between users and computer systems within 20 years. He describes his vision of *Ubiquitous Computing* as the disappearance of technology from our active awareness by its natural integration into the environment. In his eyes, modern users are no longer bound to a single computer to achieve their goals. Instead, they incorporate different devices without consciousness of the interaction with them. As an analogy, he mentions the interpretation of street signs as information perception that happens while the viewer is unconscious of the act of reading itself. The focus lays on the goal here, not the interaction.

Ubiquitous computers have three form factors: *tabs*, *pads* and *boards*

To achieve this, Weiser identified two main characteristics of ubiquitous computing devices: they need to be location-aware (while adapting their behavior accordingly) and must be available in sufficient number and various sizes. Weiser describes three kinds of ubiquitous computers with different form factors: *tabs*, *pads* and *boards*. *Tabs* act as active Post-it notes that can be attached to objects or people to achieve tracking, collecting, and reminding features. *Pads* are devices that can be used for sketching or writing

purposes just like a scrap of paper. Weiser points out that these devices do not imply possession to their users: as with a paper sheet, anyone can grab one and use it. People may spread several pads over their desks and freely arrange them the way they like to. Information sharing purposes are addressed by *boards*. These devices are large wall displays akin to black boards: the users may directly interact with the board content or simply perceive information shown on it. In Weiser's prediction, there are hundreds of tabs, ten to twenty pads and one or two boards located in a common environment.

Which technologies are necessary to realize Weiser's vision of Ubiquitous Computing? First, the devices just described should have minimal power consumption and need to be producible at low cost. Second, a flexible and high-performance networking infrastructure is needed: the data between ubiquitous computers has to be exchanged frequently and the current device location needs to be quickly updated. Finally, a software solution supporting the user interaction, providing task-dependent tools and device-independent data storage needs to be available.

High-performance and complex technology is necessary for Ubiquitous Computing

The requirements in networking infrastructure and hardware components are almost met by today's technology. A still-open issue is the availability of specialized software solutions providing support dependent on the current hardware location. The Tangible Windows concept is not able to provide a complete solution to this challenge. Instead, it picks up Weiser's vision about the natural integration of computers into the environment: tabs, pads, and boards are devices representing Tangible Windows, but modern hardware components such as tabletops also belong to the set of Tangible Windows. These devices can be positioned and arranged in flexible ways suitable for the users' current tasks.

Tangible Windows can be naturally embedded into the environment

As mentioned above, the Ubiquitous Computing devices are not personal. Users incorporate them into their work habits to accomplish their task, but there is no possession of the devices, because they are available everywhere in sufficient amount. One group of hardware components running my Tangible Windows software prototype, the Apple iPads acting as Pad Windows, belongs to the device class that Weiser named Pads. The difference between these devices

High availability of low-cost hardware is still pending

and Weiser's Pads is that they are personal: they belong to a specific person. Due to the fact that production costs for this kind of tablet computer hardware are still high, the availability of a considerable amount of non-personal computing devices is furthermore pending.

The Tangible Windows prototype supports different applications

Virtual windows on desktop computers are rectangles containing application content such as user data or interface widgets. My Tangible Windows software prototype allows users to interact with the applications similar to virtual windows on the desktop computer. But the important characteristic of a Tangible Window is its physicality: the user can pick the Window, hold it in her hands, move it around, and embed it into the environment.

1.1.2 Nomadic Operations

Tangible Windows provide a set of operations

The user of a desktop computer is provided with several operations she can apply to the application windows on her screen: they can be focused, moved, resized, minimized, opened, hidden, or closed. Tangible Windows also provide operations executable through buttons available in the graphical user interface. We denote this feature set for data exchange as *Nomadic Operations*. However, the set of operations is different from virtual windows' manipulation: since every device represents a single application window, there is no focus operation. The user just grabs the Tangible Window and is immediately able to interact with the application. There are no resizing operations since today's devices do not allow physical resizing. The feature of content moving between devices of different form factors (for example from a Pad Window to a Board Window) may be regarded as a special kind of a resizing operation. Closing and opening of Tangible Windows is likewise not feasible since it would imply disappearance from the physical world or sudden emergence.

My prototype provides two Nomadic Operations

In the following, I will describe the Nomadic Operations incorporated into my prototype that are suitable for the flexible interaction with Tangible Windows.

Receiving: The Receiving Operation is required for the selection of a Tangible Window to which application content shall be transferred. The Send Operation (explained below) requires a target selection. Each graphical user interface of a Tangible Window contains a button labeled “Receiving” which controls the activation of this operation. Thereby, users can switch the window’s receiving behavior for data on and off by just a button press.

Send: Data from one Tangible Window can be transferred to another. The application content of the copied Tangible Window is sent to the other device(s) and can be further manipulated independent from the source. Every Tangible Window set to “Receiving” obtains the sent information (while it also remains on the sending device). It is even possible to use a Tangible Window both as source and target. If the sender has an activated “Receiving” button, the sent information is duplicated on her device.

The evaluation of my prototype focused on investigating the application for Nomadic Operations and their user acceptance. I was able to show that their integration is suitable for groupware utilized in collaboration.

1.2 Iterative User-Centered Design

Software development and user interface design are processes based on many various factors. Therefore, it is impossible to design a perfectly usable system with the first attempt. To assure high usability, the system design has to be repeatedly tested and iteratively refined to achieve successive improvements over its previous versions. Nielsen [1993] described the design approach *Iterative User-Interface Design* and demonstrated its impact on the system quality with respect to usability measurements.

*Iterative
User-Centered
Design* increases
usability

The process of iterative system design is expressed through the *DIA (Design, Implement, Analyze) Cycle* shown in Figure 1.2. A development process traverses the cycle in sev-

Iterative system
development is
based on the *DIA
Cycle*

eral passes: the design stage constitutes the concept development for a system based on previous findings from the analysis phase. The next step is the implementation phase in which the system design is realized. During the first iterations the implemented systems are low-fidelity prototypes. In later stages, they become more complex until they reach their final state. The analysis phase covers evaluation methods such as literature research, surveys, and user testing. The designer is thereby able to discover known issues and problems the users experience with the system. He addresses them in the next iteration and ensures in the following evaluation sessions that the problems were indeed solved. The biggest improvements can be expected from the first iterations through the DIA Cycle since the greatest usability issues are fixed during this period.

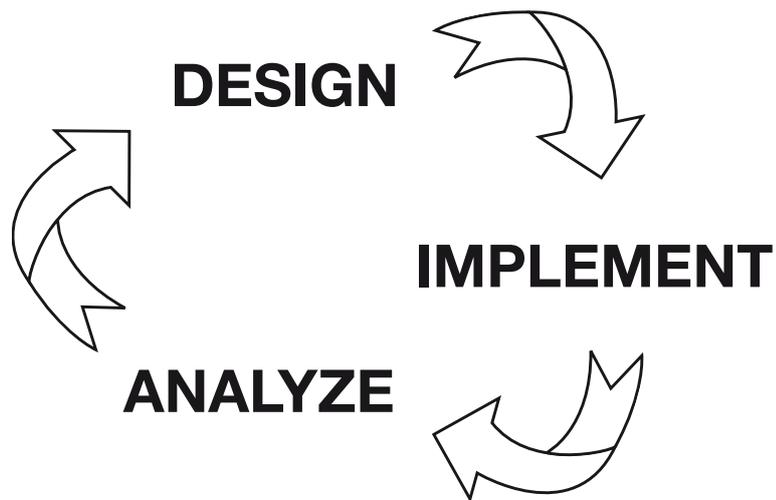


Figure 1.2: The DIA Cycle

Usability is characterized by qualitative attributes

Nielsen defines usability as a qualitative goal that is characterized by several attributes: a system is usable, if it is easy to learn, efficient, easy to remember, error-free or error-forgiving, and pleasant to use. For system usability measurement, each of the attributes can be assigned with a metric that is empirically investigated in performance tests by representative users. The attribute priorities vary depending on the evaluated project. The designer has to specify the priority of each attribute early in the development phase in order to channel his effort to the relevant design goals.

I have applied the iterative user-centered design approach to the whole development process of my software prototypes based on Tangible Windows. After an analysis of established practices in today's meetings and observations of paper-based brainstorming, I have designed and implemented prototypes that were evaluated with several user tests. Each result had great impact on the next phase and the outcomes from my system evaluation contain various improvement suggestions by users that are helpful for future iterations on the prototypes.

1.3 Chapter Overview

Chapter 1: The first chapter introduces the application area of my thesis: collaboration in meetings. I describe its relevance and common practices in group work and afterwards define the "Tangible Windows" concept accompanied by Marc Weiser's vision of "Ubiquitous Computing". The explanation of the term "Nomadic Operations" follows in the next section. The development approach "Iterative User-Centered Design" was adopted in my work on this project and is subsequently illuminated.

Chapter 2: In the second chapter, I first define the terms "collaboration" and "groupware". Then, the characteristics of collaborative work are described before I will introduce brainstorming as a special collaborative task. The important brainstorming rules as well as guidelines for the organization and structuring of a brainstorming session will be presented. Known limiting factors for brainstorming are illustrated and discussed at the end of this chapter.

Chapter 3: Chapter three presents various research projects and systems that are relevant in the context of my work. It divides into three subsections dealing with "Augmented Work Environments" including their definition, middleware solutions, and two examples for meeting software products. I explain why these projects are relevant respectively and describe similarities and distinctions between them and my groupware.

Chapter 4: The fourth chapter describes results of a survey collecting information about meetings in various business fields. The results include an overview of commonly used tools, numbers of meeting participants, and the meeting structures. In the second part of this chapter, I will discuss the outcomes from an observational study on paper-based brainstorming and its insights regarding the set-up of a brainstorming session.

Chapter 5: Chapter five describes my work on storyboards. I will first explain this prototyping approach and then present the storyboards I created. Each storyboard was presented to several users to evaluate my work's relevance for real-world settings. The results guided the following steps of my project and will be described in the remainder of this chapter.

Chapter 6: The actual system implementation is discussed in chapter six. I will present design requirements (derived from the task definition and the storyboard evaluation) on which the development of my software prototype was based. Afterwards, I will describe the networking component of the prototypes, the data model behind the implementations, and the two groupware variants targeted to iPad and Mac.

Chapter 7: Chapter seven contains the evaluation of my groupware. I conducted a qualitative user-study in a brainstorming session with seven representative participants. I will describe the study set-up and my observations during the session. After the study, the participants took part in retrospective interviews and evaluated the brainstorming and the application of my system to this meeting through a post-study questionnaire. The evaluation results will be described in the chapter conclusion.

Chapter 8: The final chapter contains a summary of my project. It presents the outcome of my work and describes how future development steps for the system might be performed. I will also present recommendations for the design of more software solutions based on the Tangible Windows concept providing Nomadic Operations.

Chapter 2

Collaborative Work

“Talent wins games, but teamwork and intelligence wins championships.”

—Michael Jordan

The interdisciplinary research field *Computer-Supported Cooperative Work* (CSCW) deals with the understanding of social interaction as well as the development and evaluation of technology supporting users in their group work. Researchers from various disciplines work together: among others, psychologists, sociologists, and computer scientists are involved. The term was first coined by Greif [1988] and Cashman referring to the goal supporting multiple individuals working together with computer systems. Although it is difficult to define unique identifying elements of this area, the term is still in use today. Bannon and Schmidt [1991] tried to specify its meaning by clarifying the individual characters in the acronym CSCW. In their eyes, “Computer Support” denotes the focus on ways to understand and improve user interaction and therefore unifies support requirements of working groups. The term “Cooperative Work” is defined as “the general and neutral designation of multiple persons working together to produce a product or service.”

CSCW investigates group work and provides supporting technology

My work on the development of a system supporting groups in their collaboration belongs to the field of CSCW.

By applying the interaction concept of Tangible Windows to meeting situations, I investigated its influences on the collaborative workflow. This leads to the question how the term *collaboration* can be outlined.

2.1 The Definition of Collaboration

Human communication is described in a four-level model

Bair [1989] provides a definition for collaboration in the context of CSCW and distinguishes collaboration from cooperation by introducing a hierarchy for these notions: he characterizes a four-level model for human communication.

With these levels, the desired amount of human interaction for a system can be specified. According to Bair, communication is characterized as the levels (1) Informing, (2) Coordinating, (3) Collaborating, and (4) Cooperating.

2.1.1 Informing

The first level (*Informing*) describes situations in which information is communicated anonymously through mass media (e.g., bulletin boards). The provider of information does not need to know its recipients and the person who wants information has to search for it.

2.1.2 Coordinating

The *Coordinating* level contains common interests of several people but excludes collective working goals. The sender(s) and receiver(s) of information know each other and are in contact. Communication is used for information and resource sharing which also includes scheduling of resources (e.g., for meeting rooms).

2.1.3 Collaborating

At the *Collaborating* level, a working relationship gets established: people participate in the same process and work together as team members on common goals, for instance the creation of a document. Each individual is likely to be in more than one team and unequal involvement in the collective effort may be present.

2.1.4 Cooperating

Cooperating denotes the ideal level: individuals' interests are deferred in favor of the team's goal and decisions are made by group consensus. Therefore, the common output is owned by all team members. Competition gets minimized and interaction occurs in high frequency. At this level, the communication between group members is always face-to-face.

2.1.5 Summary

This outcome leads to an understanding of the term *collaboration*: a common goal evolves from the working relationship between collaborators while individual interests are still present. Therefore, collaboration is the working-together in the execution of a specific action. The outcome of collaboration is a common understanding and shared knowledge between the group members. Results of the collaboration may be attributed to single participants, which leads to evaluation opportunities for individuals.

Collaboration
denotes a working
relationship with a
common goal

For this thesis, I concentrated on the study of collaborative work, because this notion fits best the situations that I investigated. Before describing the characteristics of collaboration, we must define another important term: groupware.

2.2 Groupware

As mentioned before, CSCW seeks to discover how technology can help interacting groups in their collaboration. In this context, a straightforward definition of *groupware* would simply be “software that supports groups”. A more precise definition was given by Ellis et al. [1991]. They characterize groupware as:

“computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.”

[Ellis et al., 1991, p.40]

Sketch It! belongs to the class of real-time groupware

Additional comments on this definition by the authors illustrate their focus: they explicitly exclude time-sharing and multiuser systems from groupware, because users of such systems may not share a common task. Anyway, the definition does not state that groupware users have to work simultaneously. Such systems are further specified as *real-time groupware*, as opposed to *non-real-time groupware*. The groupware based on Tangible Windows belongs to the class of real-time groupware since parallel input of several participants is possible.

Different researchers thought about the question if the terms CSCW and groupware are equivalent. Bannon and Schmidt [1991] differentiate CSCW from groupware, since they see groupware developers as technology-oriented designers who often neglect the non-technological aspects of supporting groups. In their eyes, the developers of groupware do not need to understand its application area:

“To summarize, we reject the equation of Groupware with CSCW because of its technological focus and its narrowness in the face of the multiplicity of social forms of cooperative work manifest in the world.”

[Bannon and Schmidt, 1991, p.363]

For the time being, I will use the term *groupware* whenever referring to the systems providing group support. Nevertheless, my development approach described in 1.2—“Iterative User-Centered Design” shows that the design of my software prototype was not technology-driven but included evaluation phases incorporating user feedback. Therefore, my groupware addresses a real-world setting and was tailored to support authentic tasks.

2.3 Characteristics of Collaboration

In order to design tools for supporting users with their collaborative tasks, we must define what is special about collaboration and how groupware solutions are delimited from single-user products. Tang [1991] conducted observational studies of collaborative work to investigate the interaction between collaborators in various tasks accomplished with a shared drawing surface. From his findings, he derived general design implications for collaborative software tools. In his work, he applied the research approach shown in Figure 2.1: he first observed collaborators in order to understand their workflow and then identified ways to support them. After the application of potentially supporting tools, the collaboration was again observed to investigate the tools’ impact on the workflow.

Observational studies of collaboration lead to implications for software design



Figure 2.1: Collaboration Cycle

Although his work mainly focused on collaborative drawing activities, there are findings of general interest. Participants can easily understand and work with content, only if they can follow the actual creation process. Consequently, access to other ideas should be available immediately.

Visibility of content creation is important for its interpretation

Communication increases understanding and idea quality

Additionally, he recommends assuring verbal communication opportunities for the participants during the whole collaborative process. If group members are able to exchange opinions throughout their collaboration, they can continue others' ideas and amend thoughts to existing points. Discussion phases increase the quality of the collaborative outcomes.

Groupware should provide features for parallel contribution

For a smooth collaborative workflow, it is important to provide tools that allow performing several actions in parallel. There are several known factors that can limit parallel contribution: (1) Selection, (2) Orientation, and (3) Modes: if a user selects a specific item in the collaborative working space, other group members need to recognize this action and the groupware has to assure that other participants cannot cancel the performed selection. In addition, each user should be able to freely navigate in the collaborative space (for example in a shared text document), independent from the view of other collaborators.

Modes in the collaborative software increase error probability

The design of groupware solutions should minimize the separation of actions into different modes: Tang's observations of user activities did not exhibit such segregation. When several participants in collaboration perform different actions, they need to be able to fluidly change their activities without any hindrances introduced through input modes in the software. If the tools categorize actions into modes, the user might perform an action that leads to undesired and surprising effects because of being unaware of a currently active mode. The phenomenon of *Mode Errors* has also been described by Norman [2002]: he explains that mode errors very likely occur when the system does not make its current mode visible. The user is expected to remember the established mode (sometimes even for a long time), which increases the workload and probability of error. This risk is further strengthened in groupware used by several people. Raskin [2000] describes influences of modes on the user interaction with the feature of personalization in single-user software deployed in group environments. He indicates modes in shared environments as an "invitation to disaster". If a user expects to execute a specific command, it has to be assured that the effect of her action stays the same and does not vary depending on the mode selection by other users.

If there is no way of avoiding mode integration into the groupware, visibility is the key to minimize error probability. If the currently active mode is visible to the user, it becomes less likely that she performs an unintended action. Of course, the system additionally needs to provide features for error recovery in order to reverse inadvertently executed actions.

Finally, it is important to preserve the freedom of the collaborative workflow: because people are generally very experienced in coordinating their interaction and communication, tools supporting collaboration should not impose a structure to the workflow. Tang recommends to rely on the participants' own abilities to coordinate their communication and group work.

Tools should not manage the collaborative workflow

Tang's work provides a framework of understanding collaborative work. In the following, we will look at the characteristics and specific limiting factors in the context of a typical collaborative activity: brainstorming.

2.4 Brainstorming

Brainstorming is a method for collaborative idea generation to solve a specific problem. Its goals are to increase the amount and quality of ideas generated by group members in a given time frame. Participants of a brainstorming session gather as many ideas as possible. All criticism is prohibited during the idea generation phase. Participants are encouraged to "leapfrog" off each other's ideas to extend them or to generate entirely new ideas. The more exceptional the ideas are, the better.

Brainstorming increases the performance for idea generation

The method was first coined by Osborn [1953] as a strategy for attacking a problem with a high amount of ideas. It was later extended and refined by Clark [1989]. The main characteristic of the brainstorming process lies in the emphasis on the subconscious in the human brain. Creative thinking and free associations are controlled by the subconscious, which therefore needs to be stimulated. This is accomplished by the application of four basic brainstorming

Four rules guide a brainstorming session

rules. The original formulation of these rules is the following:

1. **CRITICISM IS RULED OUT**

Adverse judgment of ideas must be withheld until later.

2. **“FREE-WHEELING” IS WELCOMED**

The wilder the idea, the better; it is easier to tame down than to think up.

3. **QUANTITY IS WANTED**

The greater the number of ideas, the more the likelihood of winners.

4. **COMBINATION AND IMPROVEMENT ARE SOUGHT**

In addition to contributing ideas of their own, participants should suggest how ideas of others can be turned into better ideas; or how two or more ideas can be joined into still another idea.

[Clark, 1989, p.70]

For their publications, Osborn and Clark studied an enormous amount of brainstorming sessions in various business fields. The common ground of these sessions is the introduction of participants into the method with the basic rules and the publication of a problem formulation to collect ideas on. Clark describes the selection of the problem formulation as essential for the success of brainstorming. He distinguishes *steam-shovel questions* and *spade questions*.

2.4.1 Problem Formulation

Steam-shovel questions are vague problem formulations

Steam-shovel questions are general problem formulations and create vague ideas by the brainstorming participants, because the question does not lead to a common understanding. Therefore, people think of different problems and their energy is not focused to the same goal. However, to brainstorm about steam-shovel questions makes sense in situations where a problem definition needs to be specified. Ideas collected on this kind of question lead to various

answers that may serve as new brainstorming questions in an additional session. For example, to first brainstorm the question "What obstacles stand in the way of increasing a company's revenue" and afterwards brainstorm one of the possible hindrances makes sense.

Spade questions are concrete problem formulations. They are limiting and as short as possible to assure a common understanding. People participating in the brainstorming session thereby focus on the same problem, which leads to specific results that can be turned into working instructions. Clark mentions an example for a spade question from the field of financial services: "How can the accounting department improve service to retail customers?" This question limits the desired answers to one department as well as one customer group, which excludes ambiguity in the formulation. Clark's investigations showed that spade questions lead to more fancy ideas (as intended by the second brainstorming rule) than steam-shovel questions, which is explained by the subconscious' structure.

Spade questions are specific and limiting

2.4.2 Participants

In addition to the importance of the problem formulation, the quality of the outcome of a brainstorming session depends on the people participating in it. Clark explains that the highest diversity is reflected in the idea collection if the brainstorming participants have different specialties, attitudes, and backgrounds. Different mindsets lead to different problem approaches and thereby increase communication between the people and the variety of their contributions.

Different backgrounds increase idea variety

Some of Clark's experiences show additional advantages when incorporating both female and male participants into the meetings. Differences in the perspectives of men and women boost the participants' productivity who therefore generate better ideas. Some companies also invite external consultants or costumers who are not familiar with the company's interests: these people add extra points of view to the problem.

2.4.3 Organization

The problem should be known ahead of the session

Clark also formulates recommendations for the organization of the brainstorming session. The invitation to a brainstorming session should already state the problem to brainstorm about. If people get invited to a meeting and do not know its subject, purposeless conjectures arise that generate rumors and prevent creative thinking before the brainstorming. Instead, the invitation to a brainstorming session must contain a clear definition of the problem. Thereby, peoples' subconscious starts working and the participants of the brainstorming session have already mulled over the problem before they meet.

Brainstorming participants should feel comfortable

The room for a brainstorming session should be carefully chosen so that the meeting participants will feel at ease to be creative in their idea generation. Therefore, Clark recommends avoiding selecting the boss's office as the location for the brainstorming session. This room might generate an uncomfortable atmosphere and have the feel of directing the future of the members. A dedicated meeting room is the best option to conduct a brainstorming session in. If there is no room ensuring the requirement to support a creative atmosphere, a good solution is to move out of the company building to an external location such as a restaurant or a hotel.

Idea recording must be efficient

The most important requirement for a brainstorming session itself is the selection of a suitable way to capture all the generated ideas. Clark describes this point in appropriate words:

“The most productive brainstorm session in history would be wasted if there was not an efficient way to capture every idea. The best ideas, in fact, quite often would be forgotten. Because of the very nature of brainstorming, the members are not trying to catch and evaluate ideas, but rush on to the next one.”

[Clark, 1989, p.96]

Common problems in brainstorming sessions are described

in the next section. Some of them arise from the fact that the session lacks a suitable recording opportunity. Therefore, the development of my groupware was conceived by the requirement to provide an efficient and suitable way of capturing the generated ideas. In general, Clark suggests to freely design the brainstorming session best suited for a given problem. The basic rules mentioned before are essential, but the structure and length of the session are freely selectable.

When the session is over, the next step is the evaluation of the collected ideas. A committee of experts usually accomplishes this. These people sort out ideas that are not practical or have already been tested and proven to be no solution to the given problem. The outcome of this judgment process is a list of the most promising ideas. Clark mentions companies that generate a list containing the top ten ideas. This list is then presented to management members deciding about the realization of specific ideas. These points get turned into tasks delegated to people or departments working on their realization.

Ideas get evaluated
after the
brainstorming
session

Although brainstorming has been proven to be a powerful method for idea generation, some factors limiting the productivity of brainstorming participants have been characterized. The following section illustrates these factors and describes their influence.

2.5 Limiting Factors for Brainstorming

Diehl and Stroebe [1987] have investigated the impact of several limiting factors of the productivity in brainstorming sessions: (1) Production Blocking, (2) Evaluation Apprehension and (3) Free Riding.

2.5.1 Production Blocking

Production Blocking accounts for the fact that only one person speaks at a time. During the expression of one partic-

Production Blocking prevents group members from generating new own ideas

ipant's ideas, others have to wait for their turn and need to hold their input until they are given the opportunity for contribution. Group members even suppress some of their ideas, because they appear less interesting at a later time. As participants have to listen to others' contributions, they are distracted from their own idea generation.

For the current contributor, the fixed time window for idea expressions can be a limiting factor as well: Dennis et al. [1990] mention *Limited Air Time* as a potential productivity hindrance. When only one person speaks at a time, there is limited room for each individual to contribute and some ideas might get lost when the speaker is interrupted.

2.5.2 Evaluation Apprehension

Evaluation Apprehension denotes the fear of negative evaluation by other group members

Evaluation Apprehension denotes a social inhibition of contributing ideas caused by fear of negative judgment by other group members. Despite the brainstorming instruction to spare any criticism, some research supports the hypothesis of people's reluctance to mention more original ideas in presence of experts in their group. However, there is disagreement about the existence and relevance of evaluation apprehension in the research community.

Nevertheless, the influence of expertise in collaboration was also shown in more recent projects: Thom-Santelli et al. [2010] observed that experts express greater feelings of ownership towards their contributions than novices do. The derived recommendation is ensuring that inexperienced group members are not excluded during collaboration and that experts do not feel threatened by others' ideas.

2.5.3 Free Riding

Free Riding is the reduced motivation for contribution

Free Riding means that people lay back and reduce their own contribution effort since other group members already mention ideas. This issue is dependent on the specific group task: it has been shown that the probability for free riding is increased if the collaborative outcome does

not give evidence for the amount of ideas contributed by each individual participant. Thus, participants are likely to work harder and the quantity of contributed ideas grows, if the collaborative outcome conveys individual contribution amounts (see Williams et al. [1981]).

2.5.4 Discussion

DiMicco et al. [2004] investigated the impact on participants' individual behavior based on the knowledge about their relative contribution amount: they conducted experiments about group discussions in which they included a shared display. The screen contained live statistics about the verbal contribution amount of each group member. The researchers found out that over-participants reacted to the statistics by restricting themselves while comparatively sparse-contributing group members did not increase their participation levels. The reason for this observation was that the under-participants were not as aware of their relative status as the over-participants. They concluded that social information may influence the individual behavior of group members and the designer has to be aware of this fact when developing tools for collaboration.

Production blocking is the most influencing factor on group productivity in brainstorming sessions, because it delays group interaction: people are distracted from developing new ideas by others' contributions. Additionally, since short-term memory of humans is strongly limited (see Dix et al. [2004]), group members are fully occupied with holding on their ideas, so there is no capacity left to think about new ideas.

Production blocking
is the main
productivity limitation
for brainstorming

To solve this problem, Diehl and Stroebe [1987] recommend offering the opportunity for writing down each individual's thoughts instead of dictating them to a participant acting as a recorder. Hymes and Olson [1992] developed a simple text editor allowing parallel user input. They achieved a significant performance increase for group brainstorming. Thereby, they also confirmed production blocking as the main limiting factor for group productivity that was eliminated by the feature of parallel user input.

Brainstorming productivity is not significantly influenced by evaluation apprehension

There is some doubt about the influence of evaluation apprehension and free riding: brainstorming is not the best task to show significant influence of evaluation apprehension on group productivity. The investigations by Diehl and Stroebe [1987] raise this issue but cannot prove its impact. Their comment on this outcome is grounded on the brainstorming instruction to free contributors from criticism. Therefore, the researchers do not see brainstorming participants as adequate candidates to test social inhibition effects.

The temptation to free ride correlates with contribution costs

Stroebe and Frey [1982] characterize the temptation of group members to free ride in an economic model. According to this model, the cost of the individual contribution (expressed, for instance, by time or effort) determines the motivation to free ride. Thus, the temptation to free ride should be greater, the higher the cost of the contribution is. Since idea generation in brainstorming sessions is very cheap with respect to this model, the free riding temptation for group members is low. Williams et al. [1981] showed that if tasks require more individual effort, considerable free riding effects are present. They conducted two experiments with shouting in which they proved that participants exerted high levels of effort if their contribution was identifiable and low levels of effort if outputs were not relatable to an individual member of the group.

The development of my software prototype was based on the paradigm avoiding the common hindrances of collaborative work described in this section. Since production blocking has been identified as the major hindrance in collaborative work, the development focus lay on simultaneous user input as the main feature addressing this issue.

Chapter 3

Related Work

“There is a single light of science, and to brighten it anywhere is to brighten it everywhere.”

—Isaac Asimov

An enormous amount of research projects and commercial groupware solutions deal with supporting and improving collaborative work in different kinds of meetings. These solutions belong to the field of Computer-Supported Cooperative Work (see chapter 2—“Collaborative Work”).

In the following, I will present a selection of projects and products that are relevant in the context of my groupware prototype. I will first describe different kinds of *Augmented Work Environments* that are sophisticated meeting rooms equipped with an extensive amount of networked hardware components. The next section describes *Middleware* solutions extending available software with additional functionality such as sharing or replication. The concluding section deals with two different *Meeting Software* products representing stand-alone software solutions.

The related projects belong to three different categories

3.1 Augmented Work Environments

Augmented Work Environments describe a set of meeting rooms that are often also labeled as “Active Spaces”, “Augmented Meeting Rooms” or simply “Augmented Environments (AE)”. Saadi Lahlou gave a definition for AE:

“AE is the product of the interplay of several layers with the physical environment: computing (ubiquitous computing), networks (pervasive networks), services, and data.”

[Lahlou, 2009, p.4]

Augmented Work Environments consist of networked devices providing data access and information exchange

Designers of Augmented Work Environments include devices such as digital whiteboards, tabletops, digital projectors, and the users’ personal devices (e.g., laptops, tablet computers or smartphones) into meeting rooms. With this hardware in place, the user interaction in meeting rooms gets “augmented” with respect to data access and information exchange. Because of an established networking infrastructure, data is no longer bound to a specific device and users are provided with an environment that serves as a flexible interface to information.

The challenges in the development of Augmented Work Environments evolve from the layers mentioned in the definition above. The devices incorporated into an Augmented Meeting Environment need to be intuitively usable and data transfer between them has to be both efficient and reliable. When the designer achieves high usability, the users’ goals attain priority over the interaction with the system (as described in 1.1.1—“The Vision of Ubiquitous Computing”).

3.1.1 i-Land

i-Land by Streit et al. [1999] is an Augmented Work Environment providing support for the collaboration within creative teams. The researchers conducted an empirical study collecting ideas to support collaborative work at the

creativity departments of five companies from the automobile and oil industry as well as advertising and consulting companies. The results of the study were a high demand for flexible furniture in the meeting rooms combined with computer support that has to be as unobtrusive as possible.

Based on this analysis the i-Land environment was designed. The researchers equipped a meeting room with specialized furniture containing computer components which they refer to as *roomware*. The initial set of roomware components consisted of an interactive electronic wall, an interactive electronic table, and chairs. These components were connected via a wireless network infrastructure coordinated by a software called BEACH.

Roomware denotes computer-equipped furniture

The electronic wall (*DynaWall*) displays information and is touch-sensitive in order to support parallel input by several people who cooperatively work on the walls' content. The researchers implemented different interaction techniques to move application windows over the wall without being physically in contact with it.

The *DynaWall* is a large touch-sensitive wall display

The interactive table was named *InteracTable* and contains a LCD projector displaying data on a touch-sensitive surface. At this table, up to six meeting participants are able to discuss and annotate information shown on it. Users can use pens for their annotations and incorporate a physical keyboard to write longer texts. Application windows on the table can be rotated and shuffled to meet the orientation of a participant's view.

The *InteracTable* supports pen interaction and orientation manipulation of its contents

The third group of roomware components are *CommChairs*, shown in Figure 3.1. There are two kinds of these chairs: one version has a docking facility offering users the opportunity to integrate their own laptops into the environment. The other variant contains an embedded tablet computer usable for annotations to information on the DynaWall.

CommChairs are equipped with tablet computers or docking facilities

An additional feature offered by the i-Land environment is the option to create shared workspaces. If meeting participants decide on forming a subgroup to work on a task independent from others, they can move *CommChairs* next to each other. The chairs are location-aware and sense the proximity between them. By creating a spatial subgroup



Figure 3.1: Picture of two meeting participants sitting in iLand's *CommChairs*, taken from [Streitz et al., 1999]

i-Land and Tangible Windows aim at flexible device usage and natural interaction

of chairs in i-Land, data between devices located on these chairs can be exchanged without being influenced by meeting participants who do not belong to the subgroup. Their results may later be presented to the whole group by showing them on the DynaWall.

The i-Land environment and the Tangible Windows concept share the focus on flexible device usage and natural user interaction. The researchers behind i-Land emphasize the need for computer support in collaborative tasks while paying particular attention to reducing the dominance of single devices.

Tangible Windows also aim at being flexible and let the users decide how to appropriately utilize them for their task.

3.1.2 iRoom

At Stanford University, Johanson et al. [2002] developed a prototype interactive workspace, called *iRoom* (see Figure 3.2). The environment is part of “The Interactive Workspaces Project” investigating interaction techniques for ubiquitous computing environments. The workspace is controlled by a software infrastructure called *iROS*.

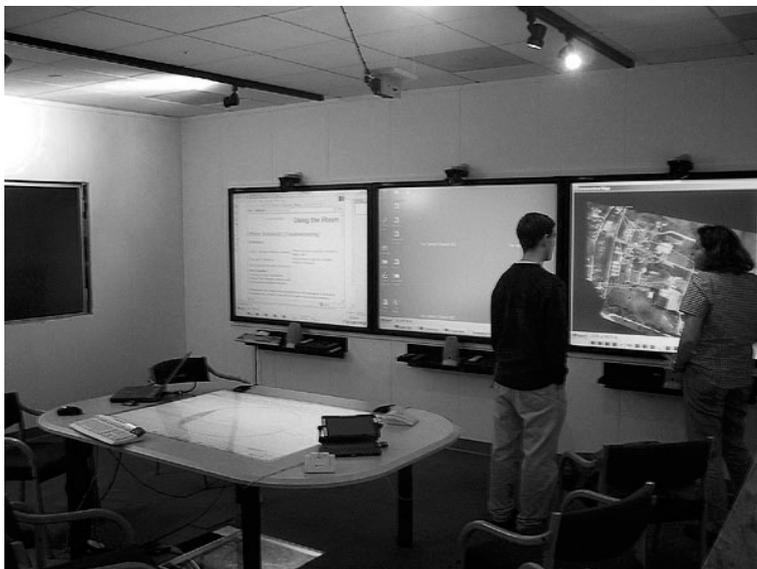


Figure 3.2: Picture of the iRoom environment, taken from [Johanson et al., 2002]

The design of iRoom was guided by principles such as the emphasis on collocated work instead of distributed usage. The user interface is kept simple and development of the software is performed as general as possible (to incorporate it into other workspaces as well).

The room was designed to emphasize collocated work

The room is equipped with three touch-sensitive displays, an additional display with pen interaction functionality, and a display-equipped table designed in a conference-table style. Furthermore, the room provides wireless LAN support and there are cameras, microphones and other interaction devices such as wireless buttons.

iRoom is controlled by a meta-operating system called *iROS*

The iROS system infrastructure is a meta-operating system. It consists of the three sub-systems Data Heap, iCrafter, and Event Heap. The *Data Heap* handles data storage and exchange within the workspace. Applications used in iRoom can place their data in an environment-associated store that additionally provides data transformation features through plug-ins. *iCrafter* is a system for service control. It contains an interface manager automatically generating and returning an interface suitable for the active services on the user's device. The communication infrastructure between iCrafter and the devices is realized with the language-independent *Event Heap* that stores and forwards "events." Events are name-type-value tuples, which can be accessed and manipulated by the running applications.

The environment was accepted by various user groups

iRoom was used in several real-world scenarios: in the environment the researchers conducted group meetings, training sessions for secondary school principals, group writing courses and other activities, most importantly brainstorming sessions by professional designers. The developers achieved positive feedback from the meeting participants and collected useful ideas for further improvement. Thereby, they were able to show the room's adequacy for different applications offering support for a variety of tasks.

The idea to develop an environment focusing on collocated work while generating benefits for various settings is similar to the Tangible Windows concept. The creative context of a brainstorming session is the setting in which my groupware was tested. However, it is not limited to this application but opens additional usage scenarios such as collaborative note taking or planning tasks.

3.1.3 The NiCE Discussion Room

The combination of traditional paper-based information, a wall display, a digital whiteboard with tangible palettes, and users' laptop computers resulted in *The NiCE Discussion Room* developed by Haller et al. [2010].

As described by Sellen and Harper [2003] in their book *the*

myth of the paperless office, paper is a widely-used medium in collaborative scenarios. It is utilized for coordination tasks, information gathering and exchange, discussion support, and information archiving. Because of the established usage of paper in meeting environments, the designers of NiCE decided to incorporate this medium into their Augmented Meeting Room: the pen-based user input on whiteboard and paper is automatically digitalized during interaction. This is accomplished by the usage of digital pens by [Anoto](http://www.anoto.com)¹. The need for human conversion of the meeting outcomes vanishes while the natural user interaction with paper remains. Other advantages such as easy replication and distribution possibilities are additionally generated.

Paper is an established medium in collaborative scenarios

The system combines private and public content generation and exchange. People are able to take private notes on paper or their laptops and decide if and what to share at a later time. Users may connect their personal computers via VGA cable to the system, there is no need for a software to be installed on their machines. The whiteboard (shown in Figure 3.3) supports tangible magnet pins, which control several overlay features of the whiteboard (overview of all sketched whiteboard canvases, displaying a connected laptop screen, and showing sketched paper content).

The system evaluation by several groups showed that most participants were satisfied with the simple data exchange in NiCE and they utilized all room features. Only half of the users were satisfied with the whiteboard interaction, because they perceived it as too large to be aware of other peoples' interaction.

Early user studies led to positive results

The opportunity for private note taking and the freedom to later share recorded ideas with other meeting participants unifies NiCE and the Tangible Windows concept. It is up to the users which of their ideas become public. The natural interaction with pens for writing tasks is also supported by my groupware. In addition, the sending feature for ideas is similar to the display of digitalized paper sketches on the whiteboard.

¹<http://www.anoto.com/the-pen-2.aspx>



Figure 3.3: A man interacting with the whiteboard in The NiCE Discussion Room. Taken from [Haller et al., 2010]

All the presented Augmented Work Environments provide sophisticated technology to support users with their collaborative tasks. The difference between these environments and Tangible Windows lies in the fact that people are still bound to a specific room and its furniture when working in Augmented Meeting Rooms while collaboration with Tangible Windows can happen nearly everywhere. Thereby, this concept creates collaboration opportunities that are independent from a dedicated environment: users can collaborate wherever and whenever they want. Tangible Windows can be carried around and are suited for ad-hoc meetings with an arbitrary count of participants.

3.2 Middleware

In this section, I describe two middleware solutions that build on top of existing applications and window managers. These systems provide support for collaborative work with features such as screen sharing and remote application control.

There are various products available providing screen sharing and input redirection functionality in networked environments. For instance *Virtual Network Computing (VNC)* is a widespread solution for desktop sharing and remote computer control. VNC is available in various *flavours*² with respect to platforms and applications. Also proprietary approaches like Microsoft's *Remote Desktop Protocol*³ provide similar features.

Many desktop sharing and input redirection solutions are available

3.2.1 ARIS

ARIS by Biehl and Bailey [2004] stands for “Application Relocator for Interactive Spaces.” This middleware manages information sharing and relocation for application windows on different devices. It enables users in meeting environments to visually relocate their applications between displays and screens, independent from resolution and scale. Also the user input can be relocated.

The relocation actions are executed through an iconic map of the meeting space (shown in Figure 3.4). The map shows all available screens, such as PDAs, laptops and tablets positioned on furniture as well as displays located at the room walls. The representation of the walls is flipped horizontally enabling the user to classify the spatial arrangement and orientation of available screens. The software runs on a middleware operating system similar to iROS (see 3.1.2—“iRoom”). Its focus lies on context-aware applications and provides support for application invocation and restore on destination devices.

Application relocation is performed on an iconic map

ARIS was developed in three iterations starting with low-fidelity prototypes and finally resulting in the solution incorporating an iconic map of the meeting room. The final version is an implementation that lets the user invoke ARIS by pressing a button in the application window bar. The iconic map gets displayed and the application window is moved to another screen in a drag-and-drop fashion.

The developers followed an iterative design process

The relocation functionality of ARIS is comparable to the

²http://ipinfo.info/html/vnc_remote_control.php

³<http://msdn.microsoft.com/library/aa383015.aspx>

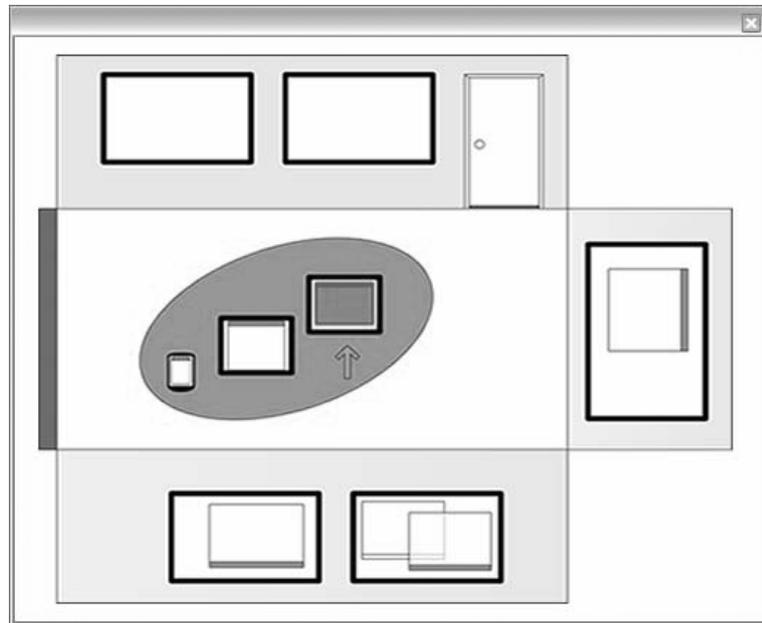


Figure 3.4: Screenshot of the ARIS relocation GUI, taken from [Biehl and Bailey, 2004]

“Send” operation of Tangible Windows. ARIS captures the application state and transfers it to the target device. My groupware already supports state transitioning but it was not incorporated into the evaluation scenario because the collaborative setting contains the danger of intensified mode errors (see section 2.3—“Characteristics of Collaboration”). These lead to confusion if other meeting participants might change one user’s application state. Therefore, my prototype confines to the exchange of documents: the recorded ideas by the participants. The simple usage of ARIS is opposed to its restriction to a specific work environment. The GUI displays the room installation and needs to be adapted if the equipment changes.

3.2.2 IMPROMPTU

Biehl et al. [2008] developed an interaction framework for multiple display environments, called *IMPROMPTU*. They conducted a user study at Microsoft to observe and im-

prove collaboration in the authentic setting of software development.

Their solution connects team members through a software framework combining private and public displays. Users are able to show and share (in order to jointly work on data with their colleagues) application windows of any existing software or place information on public screens (for discussion purposes). For the sharing feature, the researchers paid particular attention to the preservation of the interaction context (such as debug information) during input processing across multiple workstations. The user interface provides a visual representation of the group members and their activities. It consists of the Collaboration Control, the Collaborator Bar, and one or several Shared Screen Dock(s) (shown in Figure 3.5).

The interaction framework consists of three GUI components

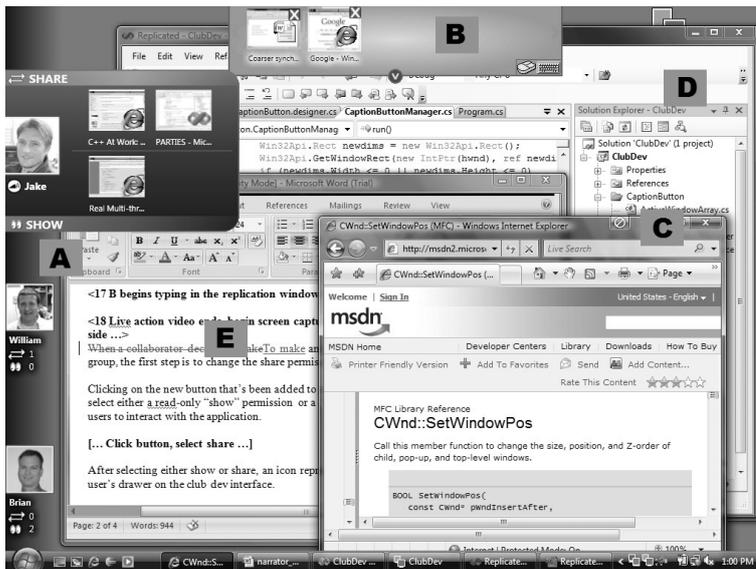


Figure 3.5: Screenshot of IMPROMPTU (taken from [Biehl et al., 2008]) showing various interface components

The *Collaboration Control* is displayed on each top-level application window and lets the user specify its sharing states. The available states are “Do not show or share,” “Show” and “Share.” When an application gets shared, other users are also able to see and interact with it, which is not possible if the application is in “Show” mode. The third state hides the application window to other group mem-

bers and is the default mode.

The *Collaborator Bar* provides a representation of each group member participating in the collaboration. It contains the windows each user made available to others. There are two rows for each user separating shared from showed application windows.

Application windows can be placed on a shared display by utilizing the *Shared Screen Dock*. Each shared display is represented by its own Shared Screen Dock. Users can drag-and-drop windows from any group member's Collaborator Bar into the dock. The window order and position can be freely adjusted.

The target area of IMPROMPTU is software development

The system has been directed to the technically-oriented special purpose of software development. The researchers report a high system acceptance by the collaborating developers who wanted to reuse the framework after the user study — especially because of the application-sharing feature. The system needs to be re-evaluated for other collaboration tasks, for example creative settings.

My groupware based on TW does not support input-redirectation, but the "Show" feature of IMPROMPTU is similar to content presentation by showing a window around the room. Shared screens are equivalent to publicly visible Board Windows that get used by meeting participants to demonstrate their ideas to others.

3.3 Meeting Software

Collaborative Work on shared files such as text documents is often accompanied by version control systems that track changes made by several editors and provide collaborators with up-to-date documents based on the work of several users. However, these systems do not provide real-time collaboration for group members who want to simultaneously modify a document.

There are many text editors such as [SubEthaEdit](#)⁴ and web-based solutions like [Google Docs](#)⁵ allowing live editing of a shared document for several users in real time.

Various solutions for real-time text editing are available

In the following, I describe two different software solutions for collaboration that are relevant in the context of my groupware, because they support creative tasks and brainstorming sessions. The first project is a system for collocated work oriented to product designers and the second software is a web-based commercial product.

3.3.1 TEAM STORM

Hailpern et al. [2007] developed a system called *TEAM STORM*. Their contribution is a demonstration of an interaction model allowing a team of product designers to create and share multiple ideas in parallel.

With *TEAM STORM*, designers create hand-drawn sketches of their design ideas on tablet computers. Their recorded ideas are organized in two workspaces: the *Personal Workspace* and the *Group Workspace*. Both are represented by windows containing miniature versions of the created sketches displayed in a spatial map (see Figure 3.6).

Sketches are organized in two workspace windows

The *Personal Workspace* contains all the sketches of a particular user. It provides zooming functionality and the small sketches can be freely positioned and scaled. The *Group Workspace* is similarly constructed, but available and identical to all group members. If users want to share their design ideas, they move the sketch representations from the *Personal Workspace* window to the *Group Workspace* window and thereby make them available to others.

The software is based on a client-server infrastructure. Every client connected to the server gets provided with access to the *Group Workspace*. If the server is connected to a large display, it visualizes the public sketches and shows manipulations performed on client machines.

The software is based on a client-server infrastructure

⁴<http://www.codingmonkeys.de/subethaedit/collaborate.html>

⁵<http://www.google.com/google-d-s/tour2.html>

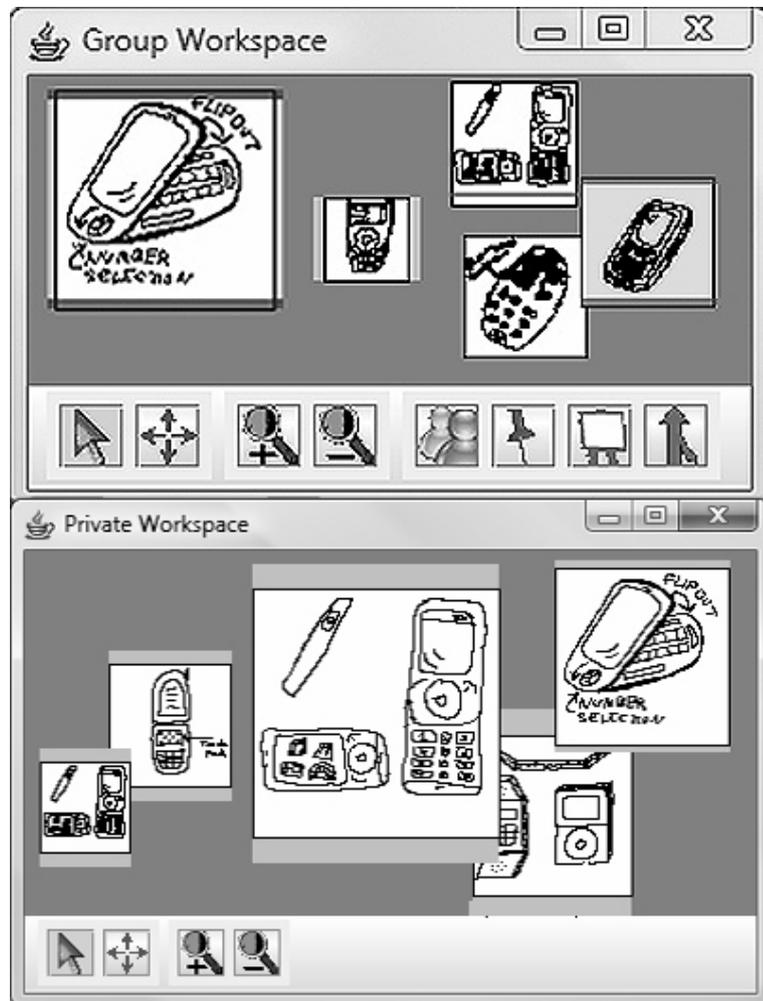


Figure 3.6: Screenshot of TEAM STORM showing the Private and Group Workspace windows. Taken from [Hailpern et al., 2007]

The researchers conducted an informal study with three groups of product designers (consisting of two and three members) working on concepts for a mobile device. Designers shared almost all of their privately created ideas to discuss them in-group. They highly appreciated the feature to visually layout the sketches on the spatial map.

The TEAM STORM system is very similar to my groupware prototype "Sketch It!": both systems provide sketching features for idea recording. TEAM STORM is explic-

itly targeted to product designers and focuses on fulfilling the principles “re-interpretation” and “reflection-on-action.” These express meeting participants’ needs to have multiple ideas immediately available and to follow their development process. Brainstorming is inherently a creative task but not limited to product designers. Additionally to sketching, my groupware contains features for text integration and editing. It may be incorporated into brainstorming sessions of any kind of business field. The Tangible Windows concept offers the freedom of choice for its usage in very different kinds of scenarios.

TEAM STORM and my prototype provide similar features but differ in their application direction

My prototype consists of two independently designed software products adapted to the special character of the window form factor (Board Window and Tab Window). It thereby provides a graphical user interface appropriately tailored to the underlying hardware while TEAM STORM runs the same interface on each device. As opposed to a single sharing opportunity (moving sketches into the Group Space), “Sketch It!” allows for various sharing variants such as public demonstration of ideas on a large screen and subgroup idea exchange between a selection of meeting participants.

3.3.2 ThinkTank

ThinkTank by [GroupSystems](http://www.groupsystems.com)⁶ is a commercial online collaboration software used by various companies and government institutions. Because of its web-based approach, it provides support for remote collaboration as well as collocated collaboration.

ThinkTank is a commercial web-based collaboration software

Among other system features, ThinkTank provides file sharing, remote presentations including user comments with votings, and brainstorming support for idea exchange with additional structuring and categorizing functionality (see Figure 3.7). The results can be turned into statistics and reporting publications for decision makers.

ThinkTank’s user group demonstrates the propagation of

⁶<http://www.groupsystems.com>

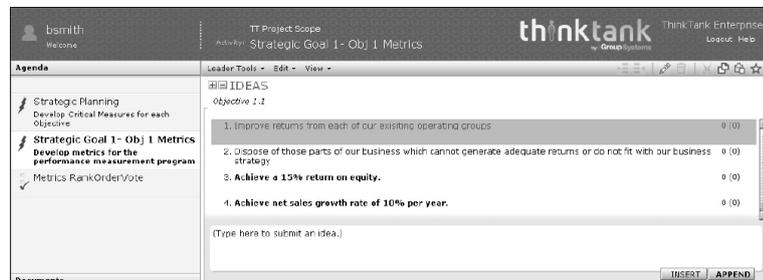


Figure 3.7: Screenshot of the participant view in a brainstorming session with ThinkTank. Image taken from [GroupSystems, 2011]

Electronic brainstorming tools are used by a variety of user groups

electronic brainstorming tools. The software is an illustration for an elaborated product also suitable for purposes beyond the actual brainstorming session. During the evaluation phase for my groupware, structuring features have been mentioned as desired improvements, which shows that subsequent work on generated ideas plays an important role for the resulting quality of a brainstorming session. Delegation functionality for working packages is already present since “Sketch It!” provides data export via email among the operation for idea sending between Tangible Windows. While ThinkTank is based on text input by its participants, my groupware combines sketched ideas and text elements.

Chapter 4

Task Definition

*“Computer science is no more about computers
than astronomy is about telescopes.”*

— Edsger Dijkstra

It needs to be specified, which application area constitutes a reasonable scenario for my groupware and which challenges are present in today’s business meetings that can be addressed by computer-based support. Furthermore, it is important to evaluate which processes within meetings need to be emphasized for successful collaboration.

As a first analysis step, I examined characteristics of meetings in various business fields with a web-based survey. With the gained results, I was able to define the application area for my groupware and specify requirements for its implementation.

The analysis specifies the application area for my groupware

Additionally, I conducted an observational study on paper-based brainstorming. Starting from its outcomes, I derived knowledge about interaction workflows in a brainstorming session and collected results for the session structure used in the design of my prototype evaluation.

4.1 Survey: Collaborative Work in Business Meetings

Guided by the work by Fowler Jr. [1995], I conducted an online survey to collect information about current practices in business meetings and the available tools. The collection of survey questions can be found in Appendix A—“Survey Questions: Collaborative Work in Business Meetings”.

By evaluating the survey outcomes I was able to get an impression of peoples’ preferences for specific devices and their experienced disadvantages in meeting situations, which shall be addressed by my groupware.

4.1.1 Participants

20 participants aged between 25 and 62 were surveyed

20 participants in total took part in my survey. The group members were explicitly selected with respect to their experience in meeting situations. 19 men and one woman answered the questions. Their ages reach from 25 to 62 with an average age of 37 (rounded).

Most of the people graduated

Five participants own a doctoral degree, six completed their master or diploma studies, seven graduated with a bachelor’s degree or a diploma degree (FH), and two participants recently ended their time at High School.

Participants come from various different business fields

16 participants are employees or freelancers. The others are students who partly graduated already. The majors of the individuals belong to various fields: economics, business administration, law, mathematics, computer science, and engineering. The kinds of businesses in which the respondents are working constitute chemicals, real estate, logistics, broadcasting and television, IT and software development, university, and also service. Examples for concerned companies are DHL and Sony Broadcast Electronics. The exact job descriptions of the participants are CEO, real estate manger, managing director, customer service operator, product manager, department manager, strategic marketing manager, graphic designer, software engineer, and research assistant.

I additionally asked for the time the participants are holding their current occupation already. The periods reach from two months to 20 years. On average, people work seven and a half years in their current job.

Participants are experienced in their jobs

4.1.2 Analysis

The questions about peoples' experience with meeting situations were divided into collections for information regarding the frequency of meeting appointments and their topics as well as group sizes and participant satisfaction in connection with them. Afterwards, I proposed questions regarding utilized tools and how helpful they are in supporting collaborative tasks.

Meeting appointments are an established part in the working schedule of the survey participants. Figure 4.1 shows how often people attended meetings during the last year. It becomes clear that the respondents are very experienced with the meeting situation.

Participants regularly attend meetings

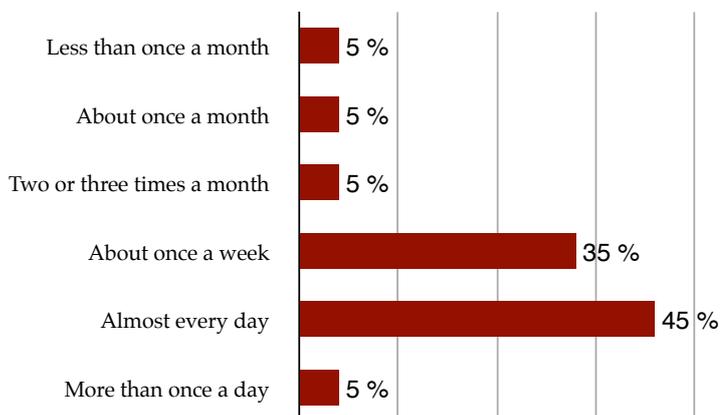


Figure 4.1: Meeting participation rates of the respondents

In order to evaluate the meeting style, I asked them to classify the amount of collaborative meetings denoting group work with a common result (as opposed to presentations that are given by a single person and do not create an out-

come generated by several people). Twelve respondents answered to participate in “many” collaborative meetings, three take part in meetings “most” of which are collaborative and “few” meetings of the remainder constitute collaborative work.

Meeting Topics

The next questions requested information about the topics and structure of the meetings.

People collaborate on a diversity of tasks and address them with different approaches

The collection of meeting topics is a varied mixture. The respondents answered (among other statements) with: discussion, organization, brainstorming, evaluation processes, and conception work. More detailed descriptions of meeting topics and their workflow are the following:

- “Short, medium and long-term strategic issues of our business partners are mainly handled in brainstorming and evaluation meetings. Real estate projects are discussed and decisions prepared in PowerPoint presentations. In addition, team building workshops are held regularly.”
- “The different meeting kinds I encounter in my job are: Customer Meetings (presentations, discussions), Business Review Meetings (data evaluation, discussions, action planning), Strategy Briefings (presentation, discussion), Sales Forecasting Meetings (data collection, discussions), Pricing Meetings (data evaluation, action planning) and Product Development Meetings (presentations, discussions).”
- “I usually prepare notes or a checklist and try to follow them or mark all checks, respectively. I also take notes for later use.”

It becomes clear that the tasks evaluation, discussion, decision-making, and development are very popular. One quarter of the participants mentioned brainstorming as an established technique to address these topics.

Meeting Structure

The meetings are very differently structured, but several similarities are present. I only mention a selection of answers to the related question here:

- “There is a person who moderates the meeting. Mostly, there are various topics, not only one focal point. All presentations are gathered by the moderating person after the meeting and are distributed via email or intranet.”
- “Usually, there is a moderator. An agenda is not posted, but mostly clear. Notes are usually taken in a standard template and afterwards distributed via email.”
- “There is a moderator who leads the discussions of the whole group. He is also taking notes on a flip chart.”
- “The project leader serves as a moderator and gives an overview of the projects’ status quo by a presentation (no agenda or slides posted ahead). All participants get asked to say a few words about their individual status quo and what they are currently working on (including possible successes or problems). The leader assigns someone to take notes. Their format is not given — the note taker has free hand to choose an appropriate format. This usually results in a standard word processor file, but OmniOutliner files are also common. The file is sent afterwards to all participants by utilizing a mailing list.”
- “Our meetings follow the topics of an agenda. Normally, they are moderated by the person who has initiated the meeting. Minutes of the meeting including responsibilities and a to-do list are being prepared after the meeting.”
- “Usually, an agenda is posted in due time prior to the meeting and a chairman moderates the meeting touching upon the agenda topics one after the other. Occasionally, urgent meetings take place in a hurry

without a fixed agenda. But in these cases, there is usually a hot and single issue to be discussed with a clear result at the end.”

- “Usually, our meetings are chaired by a single person. An agenda is issued ahead of time — or could be the latest version of an action item list that is carried through a series of meetings. Minutes are often taken and distributed (these could be spreadsheets used within the meeting if sales or sales forecasting data is being analyzed). Video conferencing and dial-in telephone conference meetings are frequently held. Face-to-face meetings are also common.”

Meeting moderation and a known agenda are very common

The descriptions of the meeting structure demonstrate the importance of moderation for the meetings. Besides, an agenda guiding the meeting process is very common. In addition, the most frequently used channel for result distribution is revealed: email. The outcomes are almost always distributed as digital documents — in standardized and non-standardized formats.

Group Size

The questions about the sizes of collaborating groups lead to interesting insights into the common numbers of participants in business meetings. Figure 4.2 visualizes the amount of the conducted meetings consisting of a specific group size.

Group sizes up to seven are very common

By combining the classes “many” and “most”, one can see that group sizes up to seven participants are very common. Meetings with group sizes between four and six represent the greatest ratio. Additionally, one participant gave the information that all of his meetings consist of exactly ten participants (represented by the “all” amount of the row for the group size ten). It is furthermore noticeable that also meetings with more than ten participants are existing: only five respondents precluded these group sizes by selecting “none”.

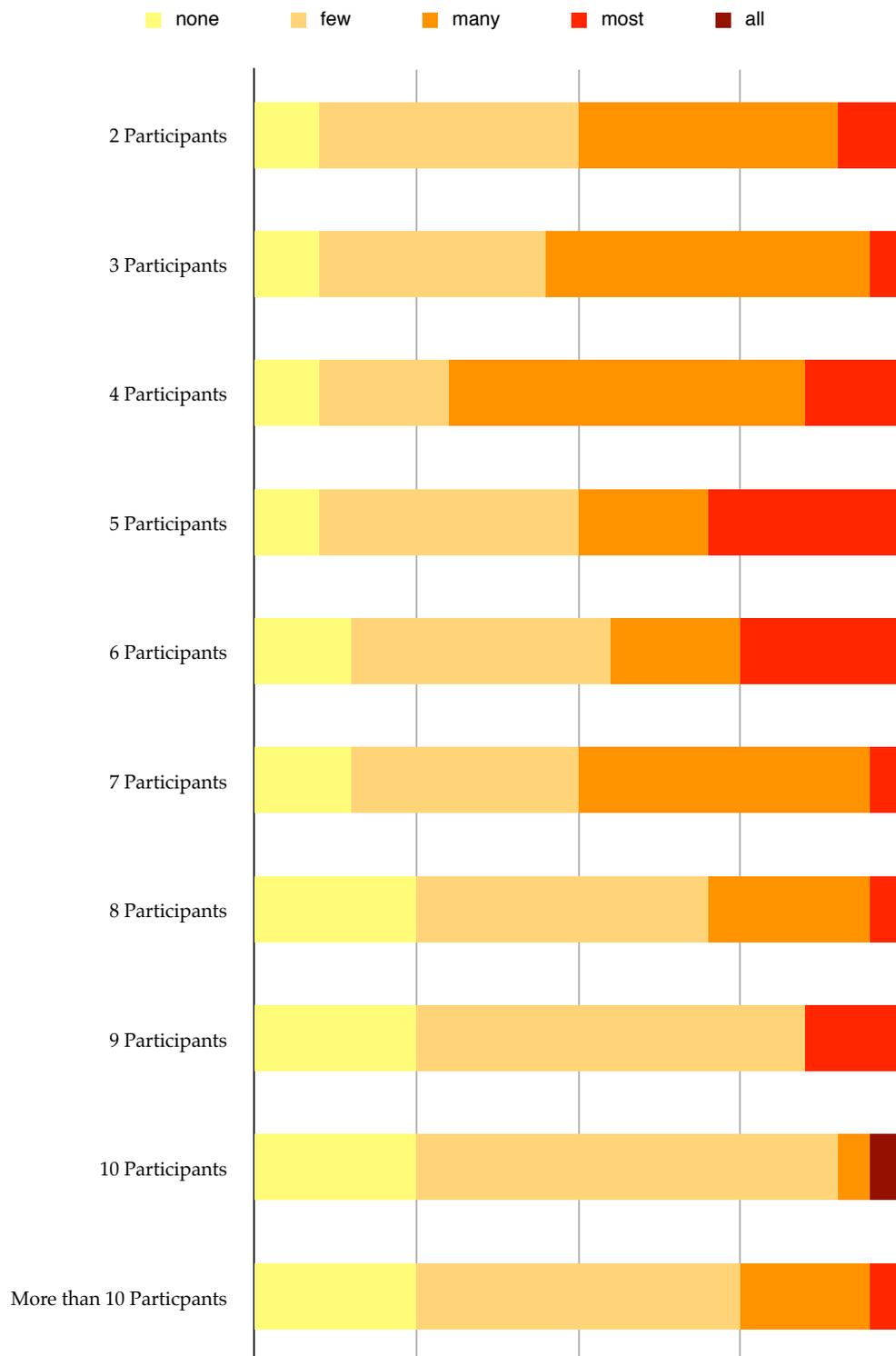


Figure 4.2: Amount of business meetings consisting of a specific group size

The satisfaction of meeting participants decreases, as group size grows

Based on the distribution of different group sizes, respondents were asked: "Given the group size, how many collaborative meetings are on average satisfying for you?"

The results are shown in Figure 4.3. It becomes clear that the satisfaction decreases, as the group size gets larger. The highest satisfaction is experienced in groups consisting of three members, but also larger group sizes (e.g., five and six) seem to be promising. One respondent even marked all meetings consisting of six and seven participants as satisfactory.

Productivity Hindrances

I wanted to investigate the reasons for dissatisfaction of the meeting participants. Therefore, I asked for the factors preventing the meetings from being successful. Some of the given answers are the following:

- "Factors could be that participants are not prepared or show no commitment. The meeting location should be well chosen and all media should be available (e.g., beamer, flip chart, etc.)."
- "Too many participants who want to show off themselves!"
- "Different education backgrounds and differences in commitment and motivation are problematic. There is no forcing for commitment by the participants and therefore mostly the same people are discussing and working out the results. That is not very effective when the group reaches a point at which no further ideas are generated."
- "Generally, the higher the number of participants the less effective the meetings get. Especially when the participants come from different groups."
 1. Lack of preparation by some participants
 2. Multiple participants are trying to talk at the same time or leaving their microphone open during live meetings

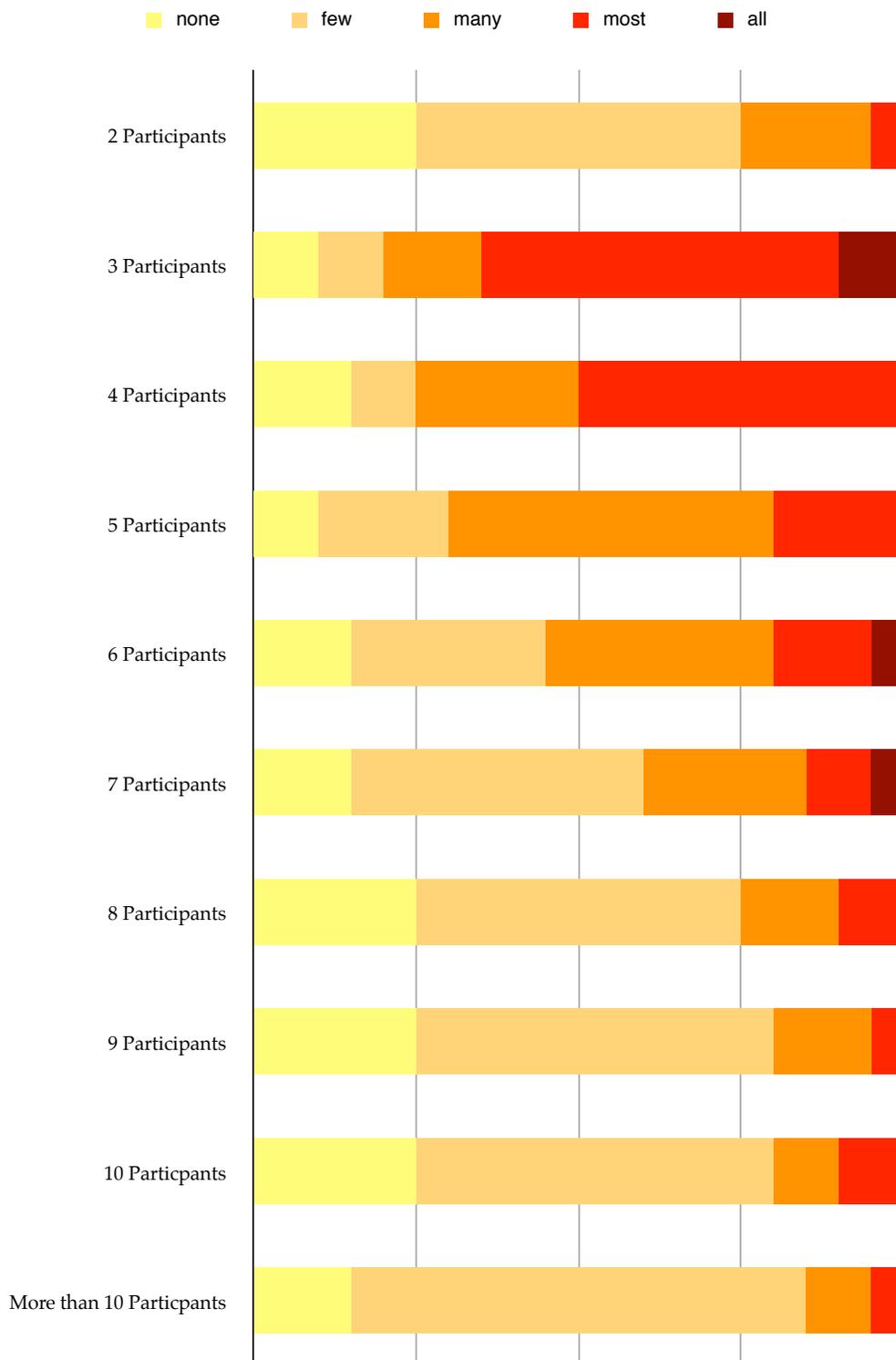


Figure 4.3: Amount of satisfying meetings related to group sizes

3. Technology problems (participants unable to share their desktop during live meetings, etc.)
4. Meeting overrunning (especially live meetings with many participants)
5. Participants are not fully focused on the meeting (e.g., writing emails at the same time or are not listening if they are participating remotely)

The main hindrances for productive meetings seem to be lack of preparation by the participants and some of the known issues of collaborative work (described in 2.5—“Limiting Factors for Brainstorming”) such as free riding and evaluation apprehension. In addition, there are technology-driven issues occurring primarily in remote collaboration.

Improvements for Collaboration

I explicitly asked the respondents to mention ideas for improving their collaboration. The most valuable answers are listed below:

- “An improvement would be finding a solution that everyone has to contribute and that there are opportunities to digitally take notes and write down own thoughts without disturbing the whole discussion flow of the group. The advantages will be that everyone’s notes can be brought together on a digital wall and the final results can be sent directly to each participant. If there are some corrections or additional notes, the information can be easily copied. It would be additionally possible to illustrate an idea or to check something on the Internet. The positive effects: time saving, equality, and results generated by the whole group (and not only input by always the same people).”
- “I think we need some tool that is able to retain the progress of our work and determines the progress for each milestone. I also would like the meeting participants to show more commitment.”

The answers demonstrate peoples' need for flexible tools encouraging meeting participants to contribute to the group work. By providing such tools, the overall productivity might get increased. In addition, sharing features for generated content seem to be promising candidates to support collaboration.

Improvements of collaboration could be flexible tools, which increase willingness for contribution

Tools

Finally, I wanted to know which tools get utilized in the meetings to support the participants in their collaboration. I presented a list of tools and devices, which had to be rated with respect to their appropriateness for keeping record of the group work. Figure 4.4 visualizes the popularity of various tools providing meeting support.

The diagram shows that classical paper-based media is utilized in every meeting and is evaluated as very helpful to support collaboration. Flip charts are used by 80 percent of the respondents, more than one half judges this tool as "helpful". Whiteboards are also quite popular: only two respondents do not find this tool in their meetings. Old-fashioned media such as blackboards or overhead projectors are rarely represented, although they are still present in some meetings.

Paper-based media is very popular

Computer support is mainly provided by laptops. Desktops and media computers are uncommon. The usage of tablet computers is yet unusual in today's meetings, 75 percent of the respondents do not utilize these devices in their meetings. In combination with laptop computers, digital projectors are often used: 15 participants rated them as "helpful" or "very helpful".

Computer support is mainly provided by laptops and digital projectors

The analysis of the popularity of tools shows that today's meetings are mainly conducted in conservative ways: the chosen devices and tools supporting collaborative work are often established parts already available for many years. Highly computer-based environments such as Augmented Meeting Spaces (see 3.1—"Augmented Work Environments") are not yet accepted by the companies of the surveyed people.

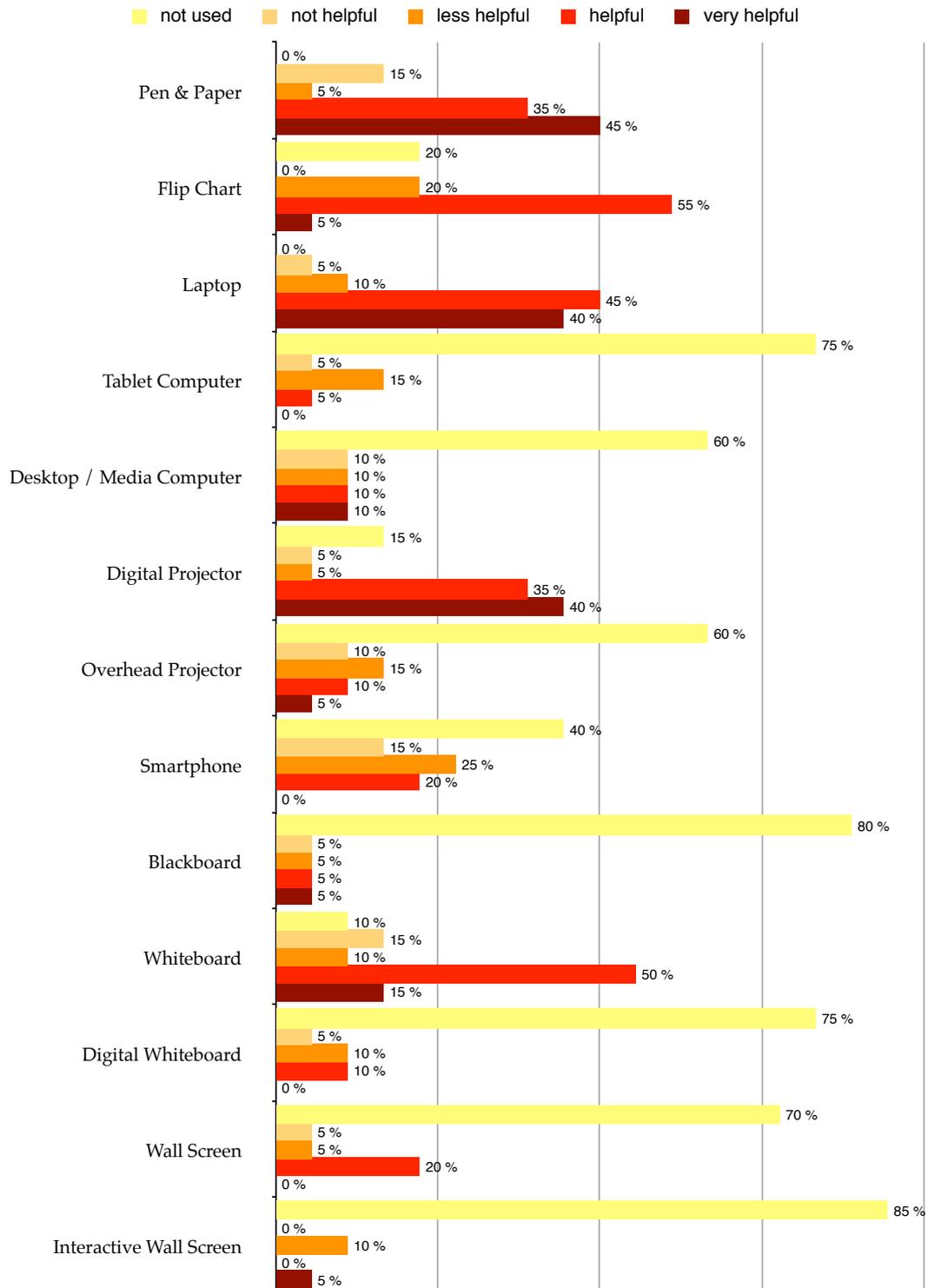


Figure 4.4: Popularity of various tools providing meeting support

Additionally to the given list, I asked the participants for further tools incorporated into their collaborative work. The answers to this question did not lead to valuable insight. Mainly, tools for remote collaboration such as web conferencing software were mentioned. Since this setting is not the focus of my work, I assume the given list of possible tools as complete.

The considered set of tools is complete

Furthermore, I wanted to examine the disadvantages of the tools rated “less helpful” and “not helpful” and asked the participants to mention reasons for inappropriateness of these tools. I present a selection of answers in the following:

- “Pen and paper are good if technical distraction is unwanted. But these tools suffer from the severe problem that my handwriting is really bad. Weeks after the meeting, I may have problems to read everything properly or to understand what I meant in that moment since I often abbreviate what I write down. Also, it is less helpful because it is in an analogue format.”
- “Laptops are not appropriate to draw ‘quick and dirty’. Nevertheless, they are necessary to look things up (but this can also distract people).”
- “Our tools are old-fashioned. When the meeting is over, the boards are usually erased and the content does no longer exist. It is also difficult (almost impossible) to share it with others, because the information does only exist on one single device and there is no technology to transfer it to other people (intern and extern).”

The answers sum up the disadvantages of paper-based media and formulate the need for tools providing quick access to the workspace for note taking in combination with features for data sharing and distribution.

Quick access to the workspace and information sharing are important

4.1.3 Results

Meeting success is highly dependent on organization

In total, the main results of the survey answers are the insight that brainstorming is a common technique utilized in today's business meetings and that the meeting success is closely related to its organization. For many respondents, the presence of a moderator and an agenda guiding the meeting process are important. Email is a heavily established distribution channel for meeting results, which leads to the insight that groupware supporting collaboration should not prevent established workflows (such as data distribution via email). Instead, it should provide a selection of features from which the groups can adequately choose according to their specific workflows.

Groupware should be easy and fun to use in order to motivate the group members to contribute and actively participate in the meetings. The group sizes should be well-chosen, because larger groups reduce user satisfaction.

The amount of polled participants is small

The results gained from the survey give an overview about characteristics of today's business meetings and provide useful insights for the design of my groupware. Anyway, the number of participants who took part in my survey is comparatively small. Although the respondents are experienced with meetings and constitute a suitable group of participants, a larger set of people needs to be asked in order to formulate more reliable results.

4.2 Paper-Based Brainstorming Session

The brainstorming session served as a source for design requirements

We conducted a paper-based brainstorming session to observe the participants' behavior and to collect first-hand impressions about the interaction between group members. Based on these observations, I was able to derive design requirements for my groupware and I also gained insight how to adequately choose the duration and structure for a brainstorming session.

The formulation of the topic for the brainstorming session

was the following:

"Imagine that your personal applications would be provided with Nomadic Operations.

What would you use them for?"

The actual outcome of the meeting was not in my focus. Therefore, the quality of generated ideas had no priority during my observations.

4.2.1 Participants

The eight participants in the brainstorming session were members of our chair. The group consisted of students and research assistants — all group members were male. Participation in the session was voluntary and there was no compensation.

Eight male members of the chair grouped as participants

4.2.2 Set-Up

The session took place in a lecture room with movable tables and chairs. The set-up was prepared in advance of the session. The participants seated themselves at a large table group consisting of a two-by-two table subset. A picture of the set-up for the session is shown in Figure 4.5.

The tools provided were post-it notes (available in various colors), DIN A3-sized writing pads as well as black-colored felt pens and ball pens. Furthermore, a whiteboard was located next to the table to be utilized for idea presentation and organization. The participants could write onto the whiteboard with felt pens.

Various tools were provided

Participants were instructed not to use any other tools than the set provided. At the beginning of the meeting, the rules were explained and the topic presented. The session structure was additionally distributed as a printout. The brainstorming itself was divided into three phases and each of



Figure 4.5: Set-up for the brainstorming session

them was scheduled to last ten minutes. A description of the three phases is given below.

Phase 1: Idea Generation

People were encouraged to write down their ideas while sitting at the table. Each idea should be written onto a single post-it note. According to the brainstorming rules (see section 2.4—“Brainstorming”), the participants were instructed to generate as many ideas as possible. They were allowed to communicate with each other, but not to comment on others’ ideas. It was prohibited to scrap any post-it notes already labeled, because we wanted to keep every generated idea. The participants could write additional notes or remarks onto the available paper sheets.

Phase 2: Content Merge

In the second phase, the group was asked to merge its ideas. Everybody should put his post-it notes onto the white-

board. Each participant was able to read the others' notes and ask questions about notes that were unclear. The creator of the concerned idea was then able to react and explain his thoughts. Redundant notes should be removed from the whiteboard to have unique ideas in the collection at the end of the session. The creation of additional notes on new post-its was possible at any time.

Phase 3: Structuring

In the concluding phase, the participants were encouraged to arrange the post-it notes on the whiteboard. In a concept-map fashion the notes should be ordered, grouped and connected by frames, connections or further illustrations. Headings or annotations could additionally be amended.

4.2.3 Observations

During the whole session, I was present in the room and took handwritten notes about my observations. Besides of reminding the group of available time, I did not interact with the participants.

At the beginning of the session, some group members started with drawings to express their thoughts. Others wrote them down in textual form. Most of the text notes were full sentences, almost no one-word note appeared. People looked around at others' notes, but for four minutes there was dead silence in the room (although communication was explicitly encouraged in the instructions). This is a common observation also mentioned by Charles Clark:

The notes were taken in textual and figurative form

"I've found that silences may naturally occur, even in the most productive session."

[Clark, 1989, p.98]

The first talking was about quality of the provided felt pens, which was not task-related but broke the silence.

Participants were very productive	Throughout the whole time, ideas were generated. Participants had almost no breaks and were very productive. At the moment when only one minute was left, I reminded the group of the remaining seconds. This led to jokes uttered by the group members who noticed that they had to hurry with the recording of their ideas.
First structuring was performed at note-adding to the whiteboard	In the second phase, the post-it notes were put onto the whiteboard. At this point, first ordering activities could be observed: some people read other's notes before they put their own contributions onto the surface. Thereby, they already put notes next to each other they perceived as related. Unexpectedly, this activity constituted a very calm situation: the reading of others' ideas took several minutes, which prevented communication. Later, complaints were mentioned regarding bad handwriting of some participants. This led to complicated readability and comprehension problems.
"Hitch-hiking" was present and led to new ideas	After three minutes, the discussion started. Different points were explained and evaluated by a subset of participants. Especially roughly sketched ideas led to discussion. While some participants were quite active in the discussion, other group members held back. They seemed to avoid criticism. Interestingly, "hitch-hiking" behavior (see section 2.4—"Brainstorming") could be observed: the reading of others' notes generated new ideas by several participants who added further post-it notes to the whiteboard.
Moving and sorting of ideas started prematurely	When only three and a half minutes were left, one participant sat down again at the table. Later on, others joined him while some group members remained discussing in front of the whiteboard. At the end of this phase, moving and sorting of notes started, which was originally intended for the last phase of the session.
Idea structuring took more time than expected	The third phase was scheduled far too short. Altogether, it took 25 minutes instead of ten. Adding of notes was still present at this stage, even at the end of the session. Some still-open questions were discussed — probably because not all notes have been read during the phase before. In addition, participants seemed to hesitate to ask about misunderstood notes. As stated in the instructions, grouping and labeling of ideas was performed: notes were moved

next to each other and were semantically ordered. At this point, parallel communication within small subgroups was observable. At the end of the session, links between groups of ideas were drawn. One of the best ideas was mentioned at this time, which seemed to be caused by a more relaxed atmosphere and hilarious mood of the participants.

One interesting insight was the fact that the adding of new ideas leads to special attention since every group member perceived the activity of putting an additional note onto the whiteboard. These ideas were definitely grasped by everybody while notes that were put simultaneously with others onto the whiteboard are more likely to be overseen.

Single adding of ideas achieves special attention

At the end of the session, more and more participants sat down again at the table — they seemed to be tired of the accomplished work.

The final idea collection of the brainstorming session is shown in Figure 4.6.

4.2.4 Results

The observations from the brainstorming session lead to useful results regarding the structure of the session and possible features for my groupware. Participant feedback after the session emphasized the importance of serial idea contribution: if a single idea is presented to the other group members, everybody pays attention to it. This leads to an improved understanding and an increased awareness of other contributions. Thereby, more exceptional ideas might get generated since people more likely build on others' thoughts.

Serialization of idea contribution improves understanding and stimulates thinking

The perception of various different recording styles for peoples' ideas leads to the result that software supporting brainstorming sessions must provide a selection of features for idea recording: people need to be able to create figurative illustrations of their ideas as well as formulate textual contributions.

Groupware for brainstorming has to provide different idea recording features



Figure 4.6: Brainstorming Result

Corrections and bad handwriting affect idea comprehension

Another result was the fact that subsequent corrections of notes impair their readability. Bad handwriting lead to delayed understanding of ideas, but crossing out of words and writing of additional text elements generated further comprehension issues.

Structuring of ideas needs much time

Therefore, software solutions should provide features for error recovery and correction tools that are both efficient and do not reduce idea readability. As mentioned before, structuring of ideas needs much time, which was an insight taken into account in the design of the system evaluation for my groupware (described in chapter 7—“Evaluation”).

The results from the observational user study are group-specific

The results from the observation are very specific to the group concerned. User interaction and the processes in brainstorming sessions might differ dependent on the kind of tasks, the topics, and the meeting participants. However, we believe that the insights give an overview about

challenges that can be addressed in the development of computer-based support for collaborating groups.

4.3 Summary

Although tablet computers are not very popular in business meetings yet, our research group believes that they are an appropriate device class to implement the groupware prototype "Sketch It!", which is based on the Tangible Windows concept. For the design, it needs to be assured that the groupware is flexible enough to meet the different requirements of interacting groups. And we consider an implementation for Apple iPads suitable for this task.

Tablet computers seem appropriate to support collaboration

As seen in the brainstorming session, both textual input and sketching functionality need to be supported. Since data sharing and exchange is a main task present in meetings, the application of Nomadic Operations (described in section 1.1.2—"Nomadic Operations") appears to be reasonable for collaborative scenarios. The tool needs to maintain the advantages of pen and paper (flexible, efficient usage and natural interaction) since this media is most established in meeting situations. Anyway, it creates disadvantages in replication and distribution opportunities that should to be overcome by the groupware.

Support of sketching and writing is a major requirement

The drawbacks of classical presentation media such as whiteboards and flip charts are also supposed to be addressed by "Sketch It!". The implementation of a software providing equivalent features to the tablet version but running on a system connected to a wall screen or a digital projector appears to be suitable for content demonstration and public interaction.

Data presentation shall be realized through a public display

The structure of a brainstorming session should be well chosen: there need to be up to seven participants to achieve a high user satisfaction. In addition, enough time for idea presentation and discussion should be available in order to enable subsequent generation of ideas and an increased understanding of contributions. To achieve best results, the meeting participants should be motivated and stimulated

Organization of the brainstorming session influences its success

by the provided tools to actively contribute to the session.

Chapter 5

Storyboards

“People think focus means saying yes to the thing you’ve got to focus on. But that’s not what it means at all. It means saying no to the hundred other good ideas that there are. You have to pick carefully.”

—Steve Jobs

Dix et al. [2004] describe *Storyboards* as a simple technique for rapid prototyping. Storyboard creation evolved as a helpful tool in the film industry for the early design of movie scenes. They show a snapshot sequence of the proposed film shots in a comic-strip style.

Storyboards are low-fidelity prototypes

In the context of Human-Computer Interaction, storyboards are hand-drawn depictions of the system used to accomplish a specific task. In the field of software development and user interface design, they are suitable tools for communicating the idea behind the system without any code implementation. The storyboards show specific interaction points of the user with the system while focusing on its main features.

Main interaction points are illustrated in the storyboards

By demonstrating the prototypes to potential users, they get a high-level overview of the system and the designer is able to early evaluate the direction of his design. The rough appearance of the storyboard sketches accompanies

the purpose of high-level system evaluation and affords the viewers to give feedback about the depicted use case and their impression of the system interaction.

The storyboard situations are meant to represent real collaborative tasks

I decided to create storyboards based on the task definitions derived from my online survey and the paper-based brainstorming session (see chapter 4—“Task Definition”) in order to gain early feedback on the prototype design. By demonstrating the storyboards, I was able to collect user opinions about the plausibility of the situations to ensure that my groupware will address real tasks occurring in meeting situations.

5.1 Situations

Six storyboards were created

I created three pairs of storyboards dealing with different collaborative tasks. Each situation is illustrated once without the usage of Tangible Windows and as another setting in which collaborators utilize Tangible Windows. The situations are based on user stories I will present in the following. The storyboards for the first situation are shown in Figure 5.1 and Figure 5.2. The whole set of storyboards can be found in Appendix C—“Storyboards”.

5.1.1 Parallel Idea Contribution

Five colleagues conduct a meeting in order to collect ideas that shall be presented to decision makers at the management department. The ideas are recorded as textual notes in a list structure.

Situation Without Tangible Windows:

A moderator keeps track of the ideas mentioned by the meeting participants on a flip chart. He writes down the contributions as they are uttered. Several people develop ideas they want to express, but since only one person speaks at a time and the moderator has to write down the ideas, users are forced to wait for each other. Therefore, they are likely to be bored and blocked in their creative

thinking during the session.

Situation With Tangible Windows:

Mike and his colleagues brainstorm with support of Tangible Windows. Each meeting participant is able to record the ideas immediately by taking notes on the tablet computer. The users do not need to take others into consideration while they contribute ideas.

During the session, a neighbor asks Mike about his opinion on the neighbor's ideas. The two easily combine their ideas in one window, which stimulates Mike's mind. He develops additional ideas while the others follow their thoughts. In the end of the session, all members' ideas are gathered in one window that serves as presentation medium to the management people.

5.1.2 Idea Access

John is a product designer. His current job is the development of an illustration proposal for a customer selling sun milk. In this context, he meets with some employees to collect illustrations to be proposed to the customer. Since the customer wants to see results as soon as possible, the group needs to hurry.

Situation Without Tangible Windows:

The team creates its illustration proposals with pen and paper. One group member, Bob, is almost finished when John asks him to present the drawing. Because of the spatial distance between John and Bob, John is not able to see the sketch in Bob's hands. The paper sheet gets passed from one group member to another until it finally reaches the opposite end of the table.

John is not confident with Bob's proposal. He knows, this illustration will not impress the customer. John crumples up the piece of paper and Bob has to start anew.

Situation With Tangible Windows:

The designers utilize Tangible Windows to develop sketches for the sun milk bottle. Because the tablet comput-

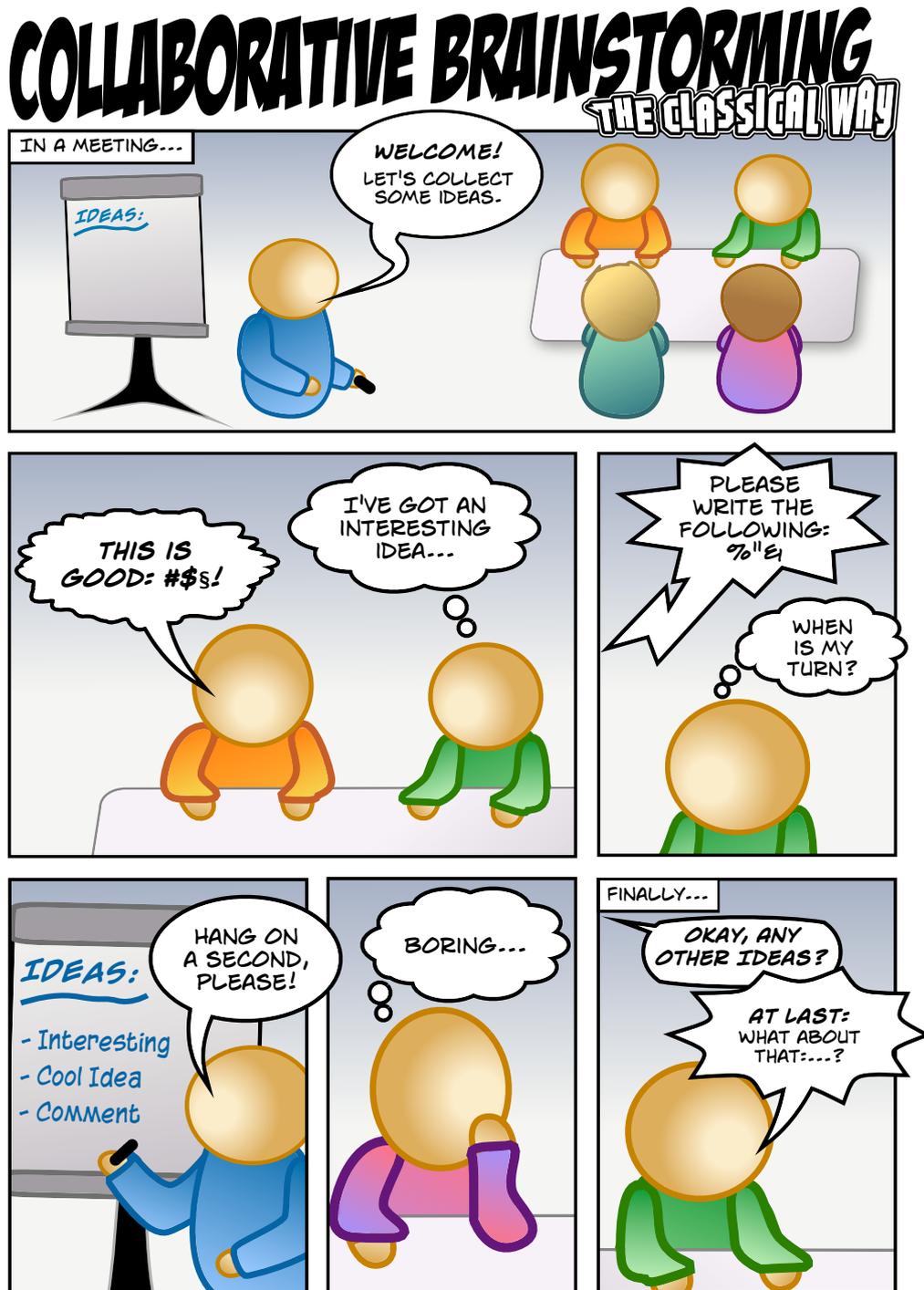


Figure 5.1: Storyboard for a classical brainstorming session

COLLABORATIVE BRAINSTORMING WITH TANGIBLE WINDOWS

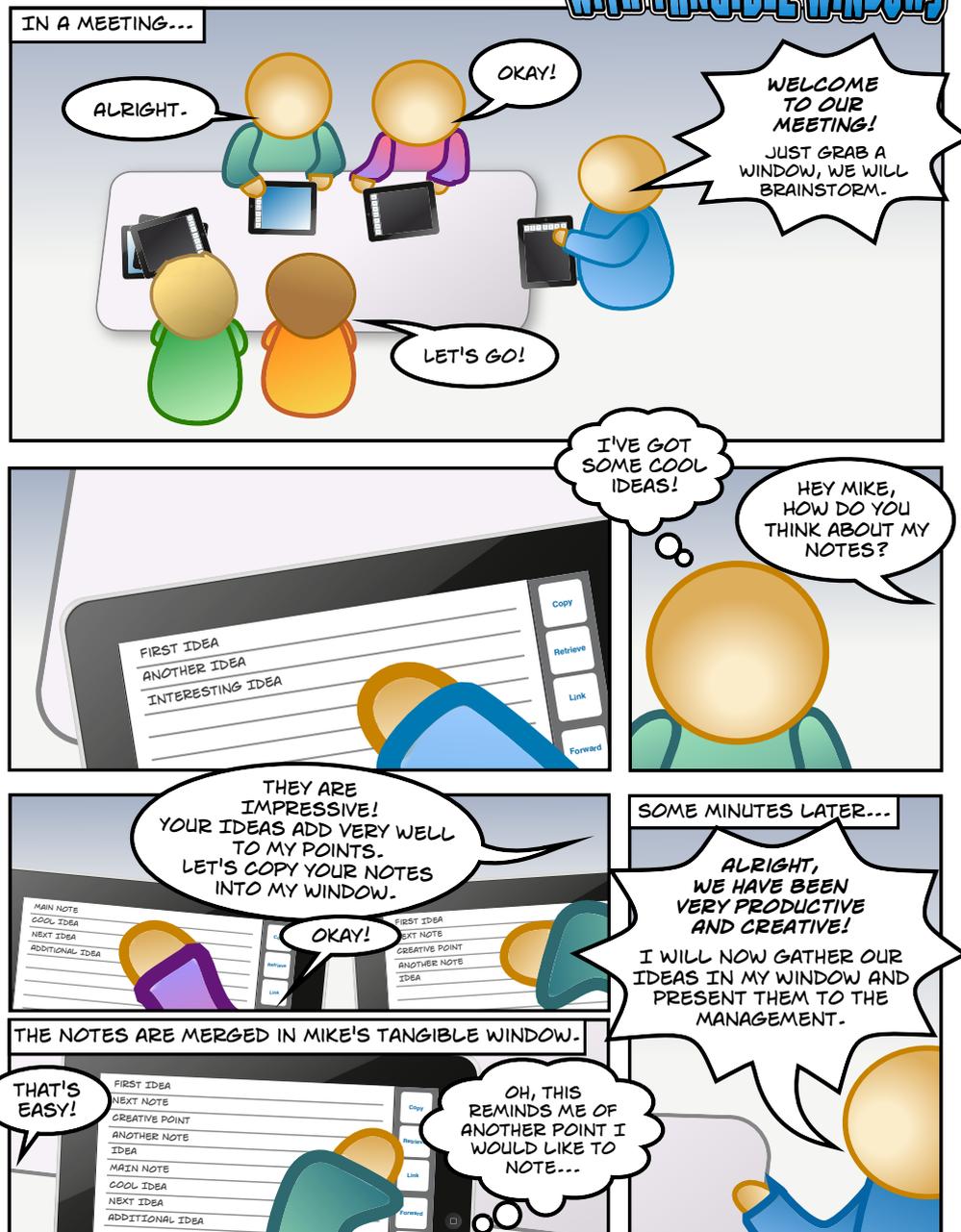


Figure 5.2: Storyboard for a brainstorming with Tangible Windows

ers are touch-sensitive, they both support input from users' fingers and pen interaction. The large LCD display serves as a Board Window to demonstrate sketches to the whole group. Because of its size, everybody can easily grasp the display content.

Bob is the first person sending his sketch to the Board Window. John criticizes the proposal but another meeting participant, Laura, has an idea: she suggests adding a face to the sun drawn by Bob. John captures the idea immediately and modifies the sketch displayed on the large screen.

Laura moves the drawing to her Tangible Window and additionally polishes the sketch by including sunglasses in the sun's face. This inspires the designer sitting next to her to combine the current sketch with his drawing of a palm tree, which quickly leads to a common illustration meeting John's vision.

5.1.3 Delegation of Work

Several employees from various different departments of the same company conduct a strategic meeting. The meeting participants come together in order to exchange their work results from the last weeks on planning a project pending for the next months.

Situation Without Tangible Windows:

Jack and his assistant Jane meet with four international colleagues. The meeting takes place in an extensively equipped environment: various tools such as a digital projector and a whiteboard with printing functionality are available as well as paper and pen-based media.

At first, one meeting participant gives a talk on a presentation he prepared in order to demonstrate the project plan guiding the future work for Jack's department. To have the important points available, Jack asks Jane to take notes with her laptop about the presentation.

The next participant gives her talk based on handwritten notes: she writes the notes onto the whiteboard. Due to the

whiteboard's built-in printer, the presenter is able to create a printout of her notes on the whiteboard.

Unfortunately, Jack has to leave early, because he needs to get his flight to another meeting in London. All he can take with him is the printout from the recently given talk. Jane's digital notes will be passed to him later via email. To have a complete overview of the project plan ideas, Jack needs to manually merge the points from the printout and Jane's notes. This constitutes unnecessary work and will consume precious time.

Situation With Tangible Windows:

The colleagues meet at a conference table. A meeting agenda gets displayed on the wall by a digital projector, so everyone can easily follow the meeting progress.

The participants make use of Tangible Windows to exchange information: the first person demonstrates his developed project plan by showing the Pad Window containing his data.

A request for data distribution gets immediately satisfied by just a button press, the chart gets sent to all other windows. Thereby, the group members immediately see the important information in front of them and are furthermore able to interact with it themselves.

One meeting participant, Kevin, gets reminded of an external project successfully completed one year ago. He asks for the project structure and gets shortly provided with the requested information by a colleague who quickly restored the data with the support of his Tangible Window.

Just like Jack from the previous story, Kevin has to leave the meeting before its ending. Since all the meeting results up to this point are stored on his Pad Window, Kevin is able to take the information with him and utilize it for an additional idea revision and extension session with his colleague Steve who he is going to meet soon at the airport.

5.2 Evaluation

I created an Ajax-enhanced HTML website for the storyboard evaluation. The URL was sent to the participants via email explaining the evaluation procedure and its goal.

5.2.1 Participants

Ten people evaluated the storyboards

The participant set for the storyboard evaluation consists of two women and eight men. Their ages reach from 25 to 56 — the average age is 34.7 years. Three participants are students, while the others hold jobs as a product manager, product specialist, technical director, customer service operators and managers. The evaluators are people who are very experienced in collaborative work. Some of them already took part in the survey about business meetings described in section 4.1—“Survey: Collaborative Work in Business Meetings”. Their judgment of my storyboards is therefore a reliable rating for the plausibility of the six situations.

5.2.2 Set-Up

The storyboards were displayed in high quality

The three situations were presented on the website including the stories described before. Each pair of storyboards was placed on the website as high-quality PNG graphics. The images could be enlarged by clicking on them to get a detailed view of the depicted sequence.

Four questions were asked about each storyboard pair

For each situation, I asked four questions to evaluate their plausibility. On both storyboards of each pair, I proposed the question: “Does the shown situation seem plausible to you?” For situations without Tangible Windows, I asked “Have you ever been in a situation similar to the one shown?” while the question about the TW-based situations was “Could you imagine to change your future behavior according to the proposed situation?”

The users gave their answers by selecting radio buttons. In

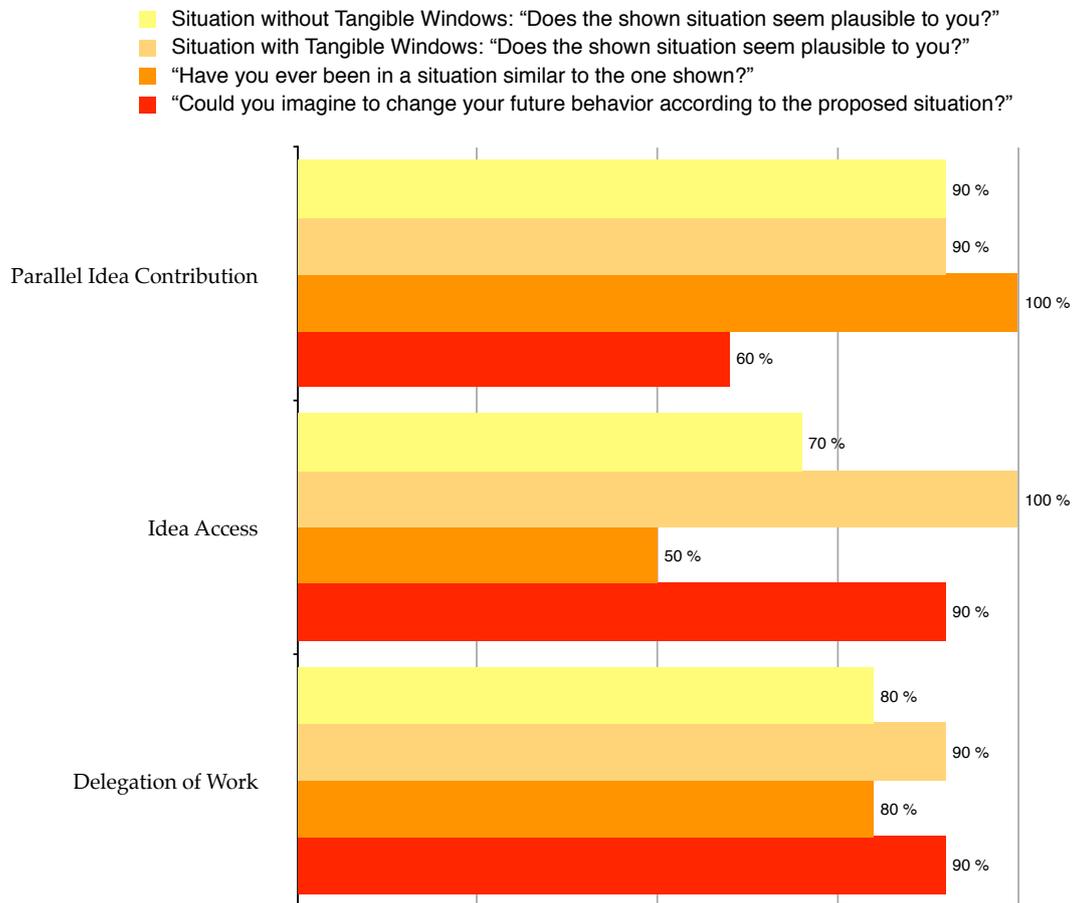


Figure 5.3: Approval ratings for the proposed questions in the storyboard evaluation

combination with personal information about age, gender and job description the form data was sent to me via email.

5.3 Results

The given answers of the participants are summarized in Figure 5.3. The diagram visualizes approval ratings for each proposed question.

The results show that a majority of the participants classified the depicted situations as plausible. Especially the situations demonstrating collaborative work with Tangible

The situations were classified as plausible

Even a unfamiliar situation represented a reasonable use case	Windows were evaluated reasonable.
	Only one half of the respondents experienced a situation similar to the second storyboard telling the user story of product designers collaborating in the creation of a sun milk illustration. Although only few people were familiar with this situation, all participants thought that the situation utilizing Tangible Windows was plausible and nine of them could imagine adapting to the illustrated TW-based workflow.
Nomadic Operations are promising features to support collaboration	The overall impression is that people liked best the exchange functionality for notes and sketches. The second and third situation focused on flexible information exchange and achieved the highest confirmation for the question if users could imagine adapting their future behavior to the depicted situation. Based on this insight, I conclude that the implementation of Nomadic Operations into the groupware is promising.
	The storyboard evaluation confirmed that the Tangible Windows concept is perceived as appropriate support for collaboration. Thereby, I was able to assure that my prototype will address real-world scenarios and will be useful for situations that take place in everyday collaborative work.

Chapter 6

Software Prototype

“Don’t be afraid to take a big step if one is indicated. You can’t cross a chasm in two small jumps.”

—David Lloyd George

Based on the analysis of meeting situations and brainstorming sessions in particular (see chapter 4—“Task Definition”), I derived a first prototype — the storyboards presented in the previous chapter.

In this chapter, I will present my second prototype, the actual software implementation of the groupware “Sketch It!”. The software is a fully functional system, which is usable in productive scenarios. At first, I specified design requirements that should be fulfilled by the software. I will describe them and afterwards show the system design with its features, before I turn to the software implementation.

In the corresponding section, I will introduce the network infrastructure of my groupware, the Nomadic Operations Framework. Then, the data model behind the software is illustrated and challenges in development of the two software variants for iPad and Mac are presented.

6.1 Design Requirements

There are six key requirements that built the foundation for the development

From literature research, my previous analysis of meetings, and the storyboard evaluation, I derived a set of design requirements for the implementation of my groupware "Sketch It!". It is based on the interaction concept "Tangible Windows" and focuses on support for collaborating collocated groups:

1. **Support drawing and typing**

As observed in the paper-based brainstorming session, participants both write down and sketch their ideas. Therefore, the software needs to support these ways of idea recording. Users will be able to take textual notes (handwritten or typed), sketch illustrations, and create combinations of both.

2. **Provide a simple user interface**

The threshold for people to utilize the software must be as low as possible. Hence, a clearly-designed graphical user interface (GUI) is necessary. People want to apply the tool effectively and will only perceive it as supportive if it is easy to use. As ideas often come to peoples' mind spontaneously and are stored in short-term memory, therefore a quick access to a workspace for idea recording is necessary.

3. **Implement various ways for data sharing**

In meetings, participants often follow established workflows to generate and share data. For instance, they distribute meeting notes via email. Tools supporting collaboration should not restrict people to completely new interaction. Instead, they are supposed to provide various ways for data sharing in order to offer users different interaction opportunities that are most suitable for their specific workflows. Besides, there must be several variants for data export to meet different application scenarios. Depending on their particular situation, people are then able to comfortably present their ideas to other members of the group and easily share information. Furthermore, it is up to the users what to share: if there are private notes not oriented to the public, users are able to

choose from their ideas a subset suitable for presentation.

4. Minimize common brainstorming hindrances

Known limiting factors for brainstorming (see section 2.5—“Limiting Factors for Brainstorming”) must be minimized by the groupware. Since production blocking is the main productivity issue in collaboration, the groupware should address this factor. The solution to overcome this problem is to support parallel user input, which is provided by my system. In addition, evaluation apprehension is addressed with giving participants the choice of deciding when and how to share their ideas by utilizing the Nomadic Operations Framework. Besides, ideas are not personalized and thereby have the property of anonymity (if intended by the collaborating group).

5. Allow correction and error recovery

The groupware should provide features for correction and error recovery such as undo and redo. There must be opportunities for users to correct erroneous content and reverse unintended actions. This way, the reliability of the system is increased. Since paper-based idea recording limits ways for content correction, the groupware has the chance of generating an added value with these features.

6. Provide fast access to information

For effective collaboration, users need to quickly access their personal ideas for modification and supplement actions. If group members want to amend information, they need to easily find and modify files they had previously worked with. Furthermore, the public workspace designed for idea presentation must provide operations for idea comparison and structuring.

6.2 System Design

The “Sketch It!” groupware consists of two implementations: one variant was implemented for Apple iPads running iOS 4.3, the other version is an application (App) for



Figure 6.1: Overview of the Sketch It! splash screen (left) and icons (iOS App icon, Mac OS X App icon, Mac OS X document icon)

Apple Mac OS X 10.6. Except for small variations, both Apps provide the same features. At first, I developed the iPad App. Afterwards, I transferred its functionality and design to the Mac App.

User interaction is possible with touch-based and pen-based input

The iPad App is controlled through touch-based input from users' fingers or a special stylus. The interaction with Sketch It! for Mac technically works through mouse-based input. However, in order to generate a natural user experience when interacting with a large screen, the software development focused on the scenario utilizing a display overlay by [SMART Technologies](http://www.smarttech.com/)¹, called SMARTBoard. This extension provides support of touch-based and pen-based interaction on large displays. Thereby, the Mac variant of my software can be controlled similarly to the iPad App. To assist this interaction, Sketch It! for Mac provides a feature to toggle the mouse cursor visibility (through a menu item and the keyboard shortcut "⌘ + Y").

Application icons generate a consistent visual appearance

For a consistent visual appearance, I created an application icon used at several places in the software. Figure 6.1 shows the splash screen displayed at launch of the iOS App, the application icons for iPad and Mac, and the document icon for stored Sketch It! files.

Screenshots of the two Sketch It! variants are shown in Figure 6.2. In the following, I will describe the main system features. Special information about the software implementation is given in an extra section.

¹<http://www.smarttech.com/us/>

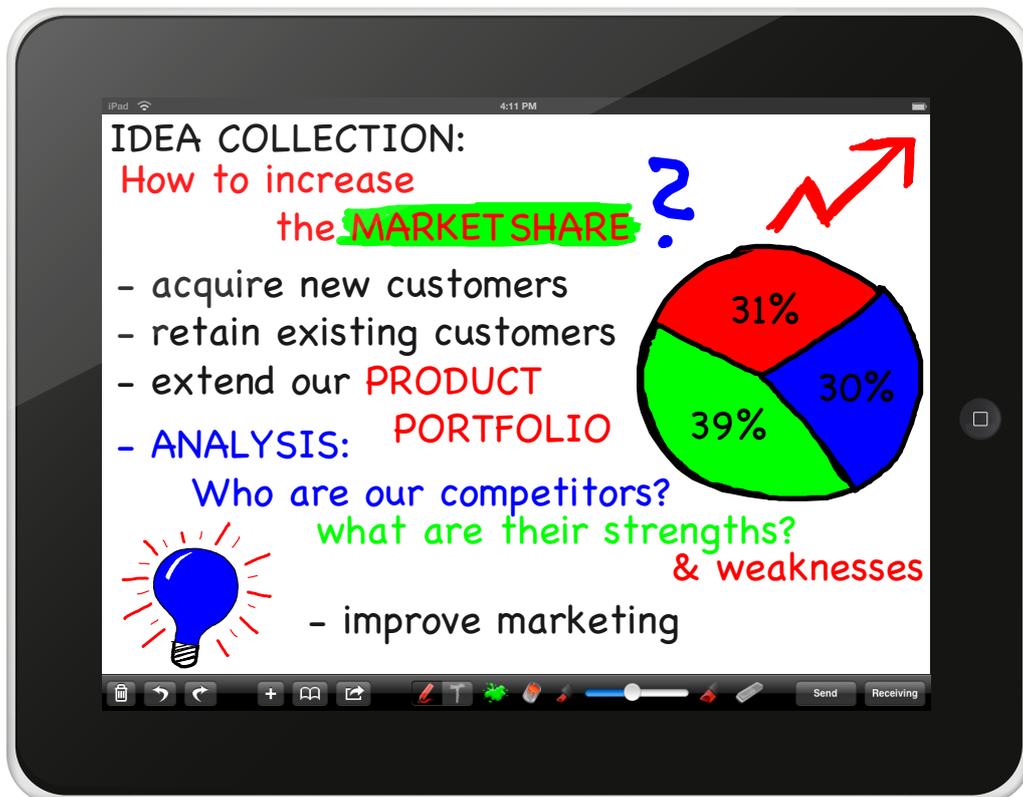
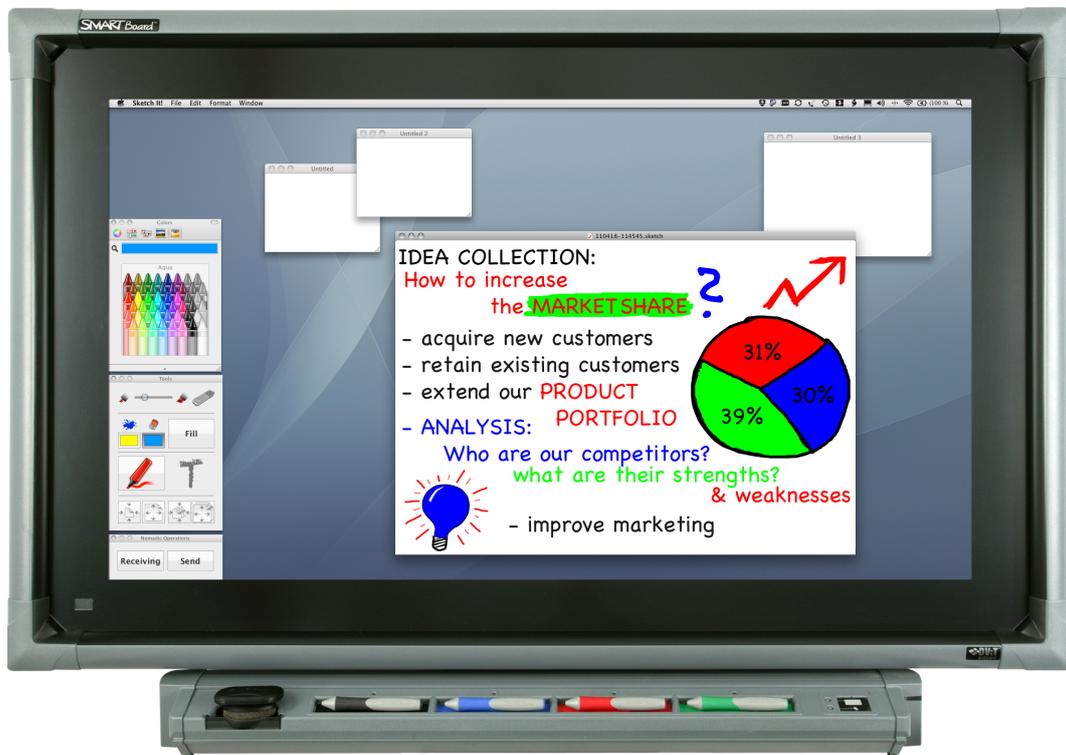


Figure 6.2: Screenshots: Sketch It! for Mac and Sketch It! for iPad

The user interface consists of a selection of tools and a drawing canvas

In both applications, the user is provided with a workspace containing a white drawing canvas and a tool collection. In the iPad version, the tools are located in a bottom-aligned bar, while Sketch It! for Mac provides freely movable tool windows containing interface widgets. In addition, the user can execute actions by selecting items from the menu bar or performing shortcuts for them.

The canvas has equal size in both applications

The canvas space is equally-sized in both versions. As shown in the screenshots, the iPad App shows exactly one canvas at a time, while the Mac version represents canvases as resizable and movable application windows.

6.2.1 Input Modes: Sketching and Typing

The two input modes are toggled with a button visualizing the current mode

The main features of Sketch It! are its two input modes: users are able to switch between a brush and a text tool, which fulfills the requirement to support idea sketching as well as typing. The input mode is selected by pressing a toggle button. The button highlights the currently selected mode to visualize it for the user. In the Mac App, the user can additionally switch the current mode with the keyboard shortcut “⌘ + T.”

The user can select brush and text colors

Independent of the current input mode, the user can specify the color of the brush and text. On the iPad, there is no standardized interface component for color selection. I therefore decided to choose the RGB color values and black as a subset from which the user can select. This set of colors should be sufficient for various purposes. The Mac variant makes the complete color spectrum available since Mac OS X provides an application-independent color widget (displayed on the top left of the screenshot in Figure 6.2). The color splash icon on the iPad visualizes the currently selected color. In the Mac App, there is a rectangular preview area displaying the current color. According to the selection, the brush color is set or it is applied to the recent text element. Figure 6.3 shows the set of interface components the user can utilize to change the input mode and specify tool properties.

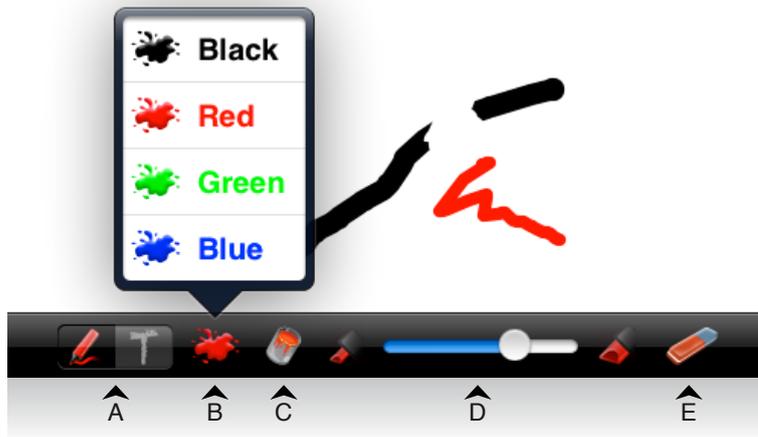


Figure 6.3: Tool selection and properties adjustment: (A) toggle button controlling sketching/typing mode, (B) brush color selection, (C) fill color selection, (D) brush size specification, (E) eraser toggle

Sketching

If the user selected the brush tool, she can create hand-drawn sketches on the canvas. Each stroke of the brush is represented by a Bézier Curve, which provides flexible and efficient manipulation techniques and is furthermore suitable for high-quality data export.

Each brush stroke is represented by a Bézier Curve

In addition to the brush color, a fill color can be specified. If the user selects a fill color in the iPad App, the recently created stroke gets filled with the selected color. The software knows the first and last points of the Bézier Path and fills its convex hull with the chosen color. It is possible to replace a path's fill color later on, by selecting a different one (for instance to distinguish several variants). In the Mac App, the path filling is applied by pressing a button labeled "Fill" placed next to the color selection.

A stroke can be filled with color

Moreover, the brush size can be controlled. This is performed with a simple slider (shown in Figure 6.3 (D)) available in both groupware implementations. Two pictograms mark the minimal and maximal brush size. The actual nu-

The brush size is controlled with a slider

meric values for the brush size reach from one to 20pt.

The brush has an eraser property

Finally, the brush can act as an eraser: a button labeled with an eraser icon (see Figure 6.3 (E)) toggles this property. If the eraser is selected, it appears colored in the GUI. Otherwise, it is greyed out. If the eraser is active, the brush color is changed to white (the selection of the actual brush color is ignored). The brush size can be changed as usual, but the path filling is deactivated since the white strokes of the brush are not supposed to be filled.

Typing

The user can freely choose the text position

The insertion of text elements is possible when the typing mode is selected. If the user touches the canvas, a dummy text element is shown. It contains the word “Text” in the currently selected color. The user can move the text and freely position it on the canvas. In the iPad version, the on-screen keyboard appears if the user releases the canvas after text positioning. Sketch It! for Mac also handles keyboard input generated after the positioning of the text element.

The text frame resizes according to its content

Both Apps show a blinking cursor at the position to where the user moved the dummy text. The characters “Text” vanish and the user can start typing. As the user types her text, the rectangular area containing the characters changes its size. Its width grows until it reaches a value of 400 pixels. The maximal height is limited to a size allowing the user to type up to five lines of text. The text element is at this height still completely visible, when the on-screen keyboard is displayed. If the user types more, a popup window is shown to inform about the exceeding of the maximal text amount for this element.

Empty text elements are removed from the canvas

Editing of the text element is completed if the user hides the on-screen keyboard on the iPad or presses the Return key on the keyboard connected to the Mac, respectively (a line break within the text element is inserted by pressing “ \backslash + Return”). If there is no single character in the text frame, it gets removed from the canvas at this point, because unintended addition of the text element can be assumed in this case.

Existent text elements can be freely moved around on the canvas. As mentioned before, the frames of text elements adapt to their content. Hence, they are as small as possible, but remain as large as necessary to display the text. Dependent on the frame rectangle, a text element recognizes touches by the user and repositions within the canvas bounds according to performed dragging gestures. Text elements stay in front of brush strokes on the canvas. This is useful, because it generates opportunities for text highlighting by drawing behind it.

The user can freely move text around on the canvas

The text itself remains editable. In the iPad App, text elements recognize a single touch and allow for text editing by setting the cursor position to the touched location in the text and displaying the on-screen keyboard. Sketch It! for Mac sets text elements to edit mode when the user performs a double-click on them.

Text elements are editable at all times

If the user wants to remove text elements from the canvas, she uses the eraser. Independent of the current input mode, the eraser button can be toggled. Whenever the eraser is activated, text elements disappear from the canvas when the user touches them.

Deletion of text elements is performed with the eraser

User Interface Improvements

I iterated several times on the design of the graphical user interface. As new features were added, the interface was evaluated by several users. Some examples of GUI elements I revised after evaluation of previous App versions are shown in Figure 6.4.

The GUI was iteratively improved

For instance, there was a toggle switch for the (de)activation of the eraser labeled with an eraser icon next to it. Although the switch was much larger than the eraser icon (and therefore easier to touch), several users tried to push the icon instead of touching the switch. At first, the icon was not touchable but only visualized the eraser state. Because of the user behavior, it was redesigned as a button that also changed the state of the toggle switch. For the final design, the toggle switch was removed, because the button with the eraser icon was expressive enough to be

The eraser icon operating as a button is the most unambiguous solution

usable. As an additional benefit, the GUI became simpler and there was more space for spatial arrangement of the remaining widgets.



Figure 6.4: Revised GUI elements: eraser toggle switch, text mode button, brush size preview image

A toggle button visualizing both input modes proved to be helpful

At first, the activation of the typing mode was realized by the implementation of a single button. In the evaluations, it turned out that users did hardly perceive the state of the button: they could not easily distinguish between activation of the typing mode and its deactivation (which implies activation of the sketching mode). Therefore, people frequently ran into mode errors and they unintentionally added text elements to the canvas. To overcome this issue, I designed a toggle button displaying both available input modes. Users testing the two iterations of the prototype perceived it as an improvement: people less often accidentally used the wrong input mode and were more aware of it than before.

The preview image visualizing the brush size was not understood

In previous versions, I integrated a preview image next to the brush size slider in order to visualize the current brush size. In my tests, it turned out that users did not understand this image. Either they did not recognize it as a brush point or they misinterpreted it as a button and tried to push it. Especially small brush sizes close to one pixel resulted in a preview image size that appeared unrecognizable. Therefore, I removed the preview from the interface since users perceived the position of the slider handle sufficient to interpret the current brush size.

By iteratively improving the user interface, it finally reached a level of usability that was sufficient to evaluate the prototype in a qualitative user study.

6.2.2 File Handling

Sketch It! for iPad features automatic file saving. After each interaction step (drawing a line with the brush, filling of paths or adding/modifying text elements), the App writes the current canvas state into a file, so the user does not need to manually keep track of her work.

Sketch It! for iPad automatically saves sketch files

New sketch files are created by just a button press: if the user presses the button labeled with a plus sign, the App creates a new file (named with a timestamp to assure uniqueness). If the user presses the button several times while the canvas is empty, only one file is generated to avoid clutter.

In addition to saving of the user's work, the App generates a thumbnail representation of the current canvas content. These images are used for preview purposes: the user is able to view and select all created sketches in a list, which consists of entries that show the thumbnails and information about the last modification date of each file. Figure 6.5 shows a screenshot of this list. It is displayed as a table view, which supports scrolling and lets the user browse through all ideas. The table rows are automatically reordered according to the modification dates of the files. Thereby, the most recently modified sketch is placed on top of the list to minimize the need for searching.

Sketch files are visualized by a preview image and the last modification date

File handling on the Mac works the classical way: as in every document-based App, each application window represents one document. In my case, the documents are sketches consisting of a canvas and its contents. Sketch It! for Mac provides file saving to the user's hard disk as sketch files with the extension ".sketch". These files can be loaded by the application and are also readable by the iPad App.

Sketch It! for Mac supports saving and loading of sketch files

6.2.3 Correction and Error Recovery

As mentioned in the fifth design requirement formulated in section 6.1—"Design Requirements", there have to be



Figure 6.5: Screenshot: selection of created sketches from a table view. Available files are visualized by a preview image and their last modification date

features for error recovery and content modification in the software. Users more likely rely on the system, if they are able to make corrections and reverse unintended actions. Some of these features have been described in the previous section. For example, the possibility to move and edit text elements as well as deleting them constitutes one set of correction functionalities. Erasing of brush strokes and the subsequent exchange of fill colors are additional modification actions.

Both Sketch It! applications provide undo and redo functionality

Furthermore, both Sketch It! variants provide undo and redo functionality. In the iPad App, the user can press two arrow-labeled buttons to step through her interaction hierarchy (see the buttons located on the left-hand side in the bar on Figure 6.5). Every insertion or removal of new elements (brush strokes/fills and text elements) is pushed on an undo stack the user can traverse. Editing of text elements or their movement is not tracked. Sketch It! for Mac provides the same feature: by executing the established keyboard shortcuts “⌘ + Z” and “⌘ + Shift + Z” or selecting the corresponding menu entries, the user can undo and

redo her actions. The menu items additionally display a short textual description of the regarding step to clarify the particular action.

In addition, the GUI of Sketch It! for iPad contains a delete button. After pressing this button, the App displays a confirmation window: the user can decide if she wants to delete the currently displayed sketch or cancel this step.

6.2.4 Organization

As described in the introduction of this chapter, both Sketch It! variants share most of their features. One difference lies in the display of sketches: while the iPad App always shows exactly one sketch, Sketch It! for Mac is able to display several sketches at a time. In addition, the sketches are resizable.

Sketch It! for Mac provides various organization features

Furthermore, in the Mac App, the sketches can be moved over the screen by touching the bar located at their top. If the particular sketch was already saved, this bar displays the filename (in combination with the document icon).

Resizing is performed in a way that is familiar to most Mac users: each window provides a handle at the bottom right-hand corner, which can be dragged to resize the window. It scales proportionally and its minimal size is limited to 280 by 180 pixels, because the sketch content is barely visible at this size. Each sketch that has a smaller size than its original dimensions is not editable. Since resizing is supposed to support organization and to provide comparison opportunities, editing of scaled-down sketches is not suitable. However, to support easier repositioning of small sketches, they can be moved by touching any point inside them: there is no need to hit the top bar as with large sketches — the user can simply drag the sketch around the screen. A double-click on a minimized sketch rescales it to its original dimensions.

Sketches are freely positionable and resizable

There are also commands to minimize and maximize a sketch. They are executable through the menu, the GUI or keyboard shortcuts. The user can scale down a sketch to its

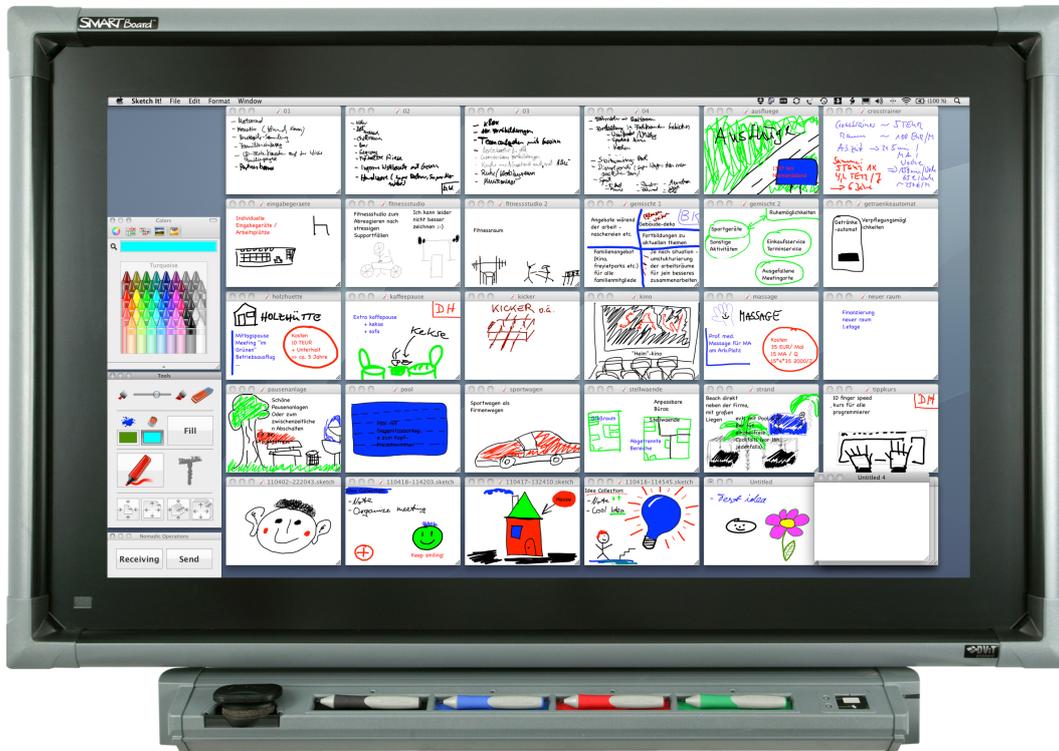


Figure 6.6: Screenshot: all sketches are minimized for comparison and sorting

Semi-automatic scaling and arrangement is provided

minimal size by pressing the keyboard shortcut “⌘ + -” and maximize it with “⌘ + +.” The sketches resize with a continuous animation to the corresponding dimensions. These commands are not limited to a single sketch: by pressing “Shift” in combination with the mentioned keyboard shortcuts, all available sketches are resized.

Additionally, minimized sketches are automatically aligned in a grid layout. An example of a sketch collection organized with this feature is shown in Figure 6.6.

These actions can also be executed with a button press: the tool window (shown in Figure 6.7) contains buttons located at the bottom that are responsible for resizing and positioning of single and all sketches.

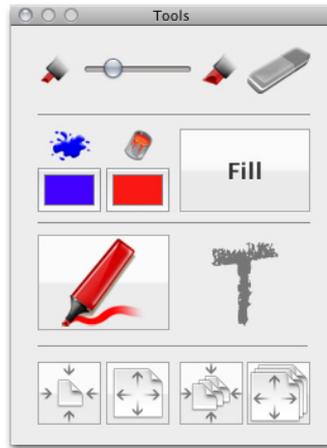


Figure 6.7: Tool window with buttons for idea organization

6.2.5 Information Exchange

The design of the Sketch It! groupware was characterized by the goal to create a prototype usable for collocated collaborative tasks. But the main focus in its development was the integration of Nomadic Operations: our research group wants to investigate if these operations provide adequate support for the scenario of collaboration.

An investigation of the appropriateness of Nomadic Operations was the main focus

Both Sketch It! applications contain functionality to receive and send sketches. As described in section 1.1.2—“Nomadic Operations”, there are two buttons in the GUIs labeled “Receiving” and “Send”. The “Receiving” button is a toggle that turns the Tangible Window into a receiver or deactivates this behavior, respectively. If the user presses the “Send” button, the current sketch is distributed to all receivers: Pad Windows store their current sketch, the canvas flashes in black color, and afterwards the received sketch fades in. The App also generates a thumbnail representation of the sketch and includes it into the collection of available sketches. Board Windows (such as the SMARTBoard) connected to a system running Sketch It! for Mac load the sent sketch and display it in full size, in front of all other sketches. The actual technical realization of this functionality is explained in the next section.

Nomadic Operations are executed with just a button press

Sketch It! for iPad provides data export via email

In addition to the implementation of Nomadic Operations, Sketch It! for iPad provides a selection of export features for sketches. It thereby supports established workflows for data distribution and sharing. The user can send sketches as an email attachment by pressing a button: if there is content on the canvas (distribution of an empty sketch is prevented), the user can send the current sketch via email. She can select from two file formats: the App supports export of Sketch It! and PDF files.

PDF files are appropriate for various applications

Since the brush strokes are represented as Bézier Curves, the generated PDF file is entirely vector-based and can be modified by various applications. Also, the text elements are editable by these. The PDF format is likewise the best choice to generate high-quality printouts. Sketch It! for Mac also provides PDF export: the user can save the current sketch as a PDF file by selecting the command from the menu or executing the keyboard shortcut “⌘ + E.”

Sketch files attached to emails can be opened on the iPad

As mentioned before, Sketch It! files are readable with both Apps. If a user receives an email attachment with the file extension “.sketch”, Sketch It! for iPad recognizes this file as a dedicated document. The user can open the attachment by touching it in the Mail App on the iPad. The same interaction works on systems running Mac OS X and Sketch It! for Mac.

Comparing the design requirements listed in section 6.1—“Design Requirements” and the system design described on the last pages, all design requirements were addressed during development of the prototype. The system evaluation described in chapter 7—“Evaluation” will show how properly they were met and which improvements may be applied in further development.

6.3 System Implementation

The Sketch It! groupware was implemented in Objective-C 2.0 utilizing the APIs [Cocoa](http://developer.apple.com/technologies/mac/cocoa.html)² and [Cocoa Touch](http://developer.apple.com/technologies/ios/cocoa-touch.html)³ by Apple.

²<http://developer.apple.com/technologies/mac/cocoa.html>

³<http://developer.apple.com/technologies/ios/cocoa-touch.html>

In the following, I will illustrate key components of the software and a selection of implementation challenges I encountered during development.

6.3.1 Nomadic Operations Framework

The framework realizing the Nomadic Operations used in Sketch It! is an implementation of networking features based on the Open Source technology [Bonjour](#)⁴ by Apple. This technology enables automatic discovery of devices and services within IP networks.

The Nomadic Operations framework is based on Bonjour

The current framework is a client-server infrastructure consisting of Nomadic Clients and a Nomadic Server. In the future, the framework will be converted into a peer-to-peer architecture, which is independent from a dedicated server.

Hence, every Tangible Window running Sketch It! is a client of the Nomadic Server (“NAServer”). All devices establish a wireless network connection to a shared access point. The system running a NAServer recognizes the clients available in the wireless LAN and uniquely identifies them by an IP address (including port number). The server receives and handles commands from clients such as the activation and deactivation of their “Receiving” mode and the sending of data. A dedicated object called “NAS-tate” represents this data.

The clients are connected through a wireless network

The implementation of the Nomadic Client “NAClient” utilizes the delegation design pattern. Every object representing a client must implement two delegation methods of the NAClient class. Besides these methods, the client objects do not have to implement further functionality. The necessary operations for network communication are encapsulated in the framework classes.

Objects acting as a client of the Nomadic Server must implement two delegation methods

- (void) client: (NAClient *) theClient
didAcceptState: (NASState *) theState

This method is called when the client receives a

⁴<http://developer.apple.com/opensource/>

NASState object from the server. The state may be an object representing the current application state (e.g., currently selected tools or GUI settings) or document data such as sketches in my case.

The implementation of Sketch It! for iPad also supports transfer of the application state consisting of the currently selected input mode and brush properties. However, this feature was deactivated for the evaluation of the system in the context of the brainstorming scenario.

- (NASState *)stateToBeSendByClient:
(NAClient *)theClient

This delegation method is called when the user performs a “Send” operation. The method implementation must generate a NASState object encapsulating the data to be sent. This data object is the return value of the method and is transferred to the Nomadic Server by the NAClient object of the framework.

6.3.2 Class Model

The class model was designed to guarantee data exchange between both Sketch It! variants

The Sketch It! software consists of several model classes. In order to guarantee compatibility of exchanged data between the iPad and the Mac version, I had to design a class hierarchy that is completely independent from API specifics of Cocoa and Cocoa Touch. Classes and data types only available on one of the platforms could not be used. Although there are some classes that seem to be equivalent (such as the different color representations UIColor and NSColor), it turned out that I had to restrict to finding the least common denominator of both APIs, which meant to confine to basic data types.

Figure 6.8 shows a simplified diagram of the final class model containing the model classes and their most important properties. Additional variables as well as method definitions are omitted. In the following, I will describe each of the classes and their implementations:

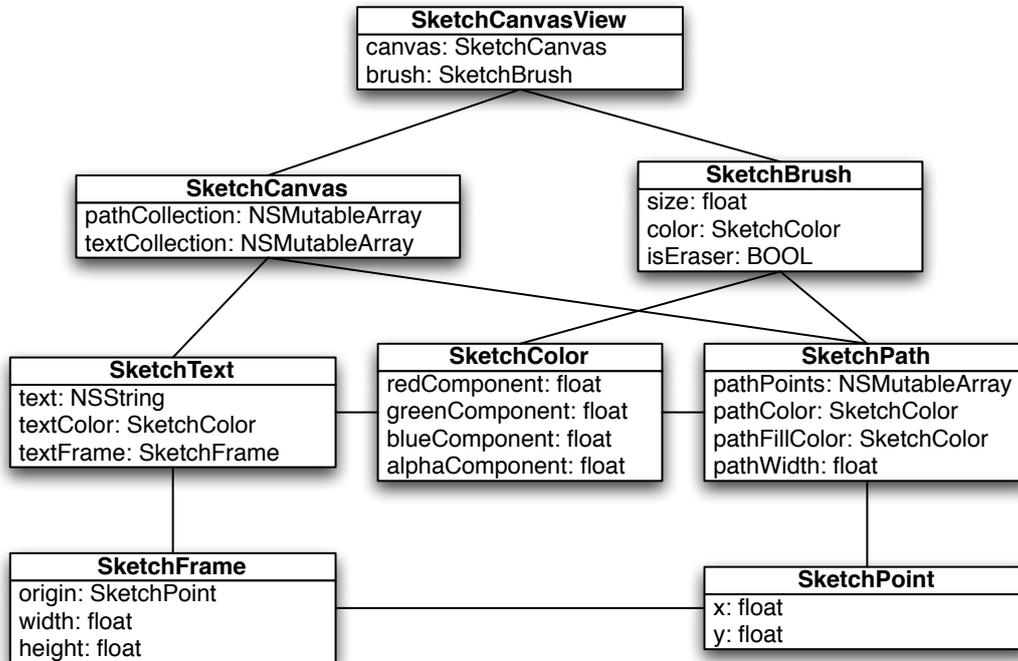


Figure 6.8: Class diagram showing the important model classes of Sketch It! with their relations

- **SketchCanvasView**

This class is the root object of the hierarchy and is the only class in the diagram being platform-dependent. It is a subclass of `UIView` or `NSView`. The according object is a view acting as a container for the canvas element (which is an object of the class `SketchCanvas`) and holds a variable of a `SketchBrush` object keeping the current brush properties.

- **SketchCanvas**

The canvas class is a wrapper for path and text elements. It provides two properties realized as mutable arrays that contain brush strokes and text elements. These elements are objects of the `SketchText` and `SketchPath` classes. The canvas class provides several methods for manipulating the arrays and obtaining information about their elements. In addition, the `SketchCanvas` implements coding and encoding methods to generate a data representation of the sketch and decode files to be loaded.

- **SketchBrush**

The brush class holds variables for the brush properties. The size is stored as a float value, the eraser property is represented by a `BOOL`, and the current brush color is specified with a `SketchColor` object.

- **SketchText**

Text elements are represented by this class. It contains properties for the character string, its color, and the rectangle framing the text. Besides the color, the appearance of text elements is fixed: they have a constant size of 50.0 points and the text is set with the font “Chalkboard.” The visual appearance of this font supports the brainstorming character of the application. Depending on which platform implements this class, related objects can call a method that converts `SketchText` elements into either a `UITextView` or a `NSTextField`, which are the actual objects used for the typing tools.

- **SketchPath**

Brush strokes on the canvas are realized by the `SketchPath` class. It contains a mutable array storing the path points (`SketchPoint` objects) and variables for the path width as well as the path color and fill color, which are realized as `SketchColor` objects. As mentioned before, every brush stroke is a Bézier Path. The `SketchPath` class provides methods to generate an `UIBezierPath` in Sketch It! for iPad and a `NSBezierPath` object for the Mac implementation.

- **SketchColor**

The `SketchColor` class is a wrapper for the color components red, green, blue, and alpha — each represented by a float value. Objects of this class can be initialized with either a `UIColor` or a `NSColor`, according to the platform. It also provides methods to convert a `SketchColor` object into the platform-specific types.

- **SketchFrame**

`SketchFrames` are used by `SketchText` objects. In order to model the frame of a text element, it is represented by a custom object storing the origin

(SketchPoint) and two float values for the frame width and height.

- **SketchPoint**

This class is a simple wrapper for X and Y coordinate values of a point on the drawing canvas. SketchPoint objects are used by SketchFrames and SketchPaths to define the frame origin and the points of a brush stroke.

6.3.3 Implementation Challenges

Because of the platform-independent design of the Sketch It! model classes described in the previous section, the data representation of sketches is compatible with both groupware implementations. However, there are additional challenges I had to deal with. These challenges are discussed in the remainder of this chapter.

Coordinate Systems

The coordinate systems for views in iOS and Mac OS X are different. In iOS, the default coordinate system has its origin in the upper left, which means that positive values extend down and to the right from it. The origin of the coordinate system in Mac OS X is located at the lower left of a view. Hence, although sketch files are exchangeable between the two Sketch It! implementations, they appeared upside down when loaded into the App that was not the creator of the file.

To solve this issue, I implemented a flag variable in the SketchCanvas class. This BOOL value is set at initialization of the canvas object indicating if the canvas object was created in Sketch It! for iPad or on the Mac. When the corresponding sketch file is read, the App checks this flag. If it is not consistent with the current platform, a method is executed to invert the coordinate system. If the file gets modified on the second platform, the flag is updated.

The coordinate system for views in iOS and Mac OS X are different

A flag indicates the creation platform of a sketch

The locations of text elements and path points get converted into the corresponding coordinate system

Unfortunately, on the iOS platform, it is technically not feasible to call a method that takes care of the “flip” of the coordinate system. Because of this reason, a custom adaptation of the canvas content was necessary. To be specific, the method cycles through the paths in the canvas’ path-Collection array and converts all Y-coordinates of the path points with multiplying them by (-1) and moving them upwards/downwards by 704 pixels (which is the canvas height). An equivalent transformation is applied to the text elements on the canvas: the Y-coordinate of the origin of each SketchText is modified in the same way as the path points.

Text Handling

As described before, the frame of text elements adapts to its enclosed character string. Thereby, it occupies as little space as necessary and the user can move text elements all the way to the canvas corners.

Resizing of a text frame is performed in delegation methods of the text elements

The resizing of a text frame is performed in delegation methods of the UITextView and NSTextField objects representing text elements. The UITextView class provides delegation methods that are implemented by the SketchCanvasView class. There are four methods called: at the beginning of text editing (triggered by a touch into the text element), the point at which the character string is about to be changed, the actual text modification (insertion or deletion of characters), and the ending of editing.

A check for the maximal frame size is performed before character insertion

The method called before the actual application of text modifications checks if the text exceeds the size limits for the text frame: it calculates the dimensions of the new text containing the character to be inserted and compares it with the maximal frame height of five text lines. If this size is exceeded, the character insertion is prevented and a popup window is shown.

If the character string fits into the frame, the rectangle size gets adapted

If the maximal frame height is not exceeded, the character is inserted and the frame grows in its width by 70 pixels per each new character to assure visibility for wide characters. At text deletion, its width shrinks if there is only one text

line left that is shorter than the current frame width. Its height is adapted if the new character generates a line break and exceeds the current frame height. The minimal size of a text frame is set to 120 by 60 pixels.

In addition to the frame resizing, each adaptation must be performed within the canvas bounds. Therefore, not only the frame size is changed but also the position of the text element on the canvas is modified if the frame exceeds the canvas bounds.

The reason for limiting the height of text elements arises from the fact that there is very limited space on the iPad screen if the on-screen keyboard is displayed. The SketchCanvasView needs to be moved according to the position of a text element that is about to be edited in order to keep it visible to the user. It is technically not feasible to calculate the cursor position within a text element in its coordinates. If this would be possible, the view could be moved such that the current cursor position is vertically centered above the keyboard. Instead, I had to assure that the height of a text element does not grow larger than the remaining screen space at keyboard display.

The height of text elements is limited because of the on-screen keyboard in Sketch It! for iPad

If the text element that is edited would be covered by the keyboard, the view containing the canvas slides into a new position when the on-screen keyboard is displayed. There are three cases that have to be considered: either the frame of the text element is positioned in the canvas area above the on-screen keyboard or it is partially or completely covered. An illustration of the described situations is shown in Figure 6.9. In the two latter cases, the slide distance for the view is calculated in the UITextView delegation method that is called at the beginning of text editing.

To assure visibility of the text element, the SketchCanvasView moves

The SketchCanvasView moves continuously into a position that displays the according text element vertically centered in the screen area above the keyboard. If the frame size changes, the same behavior is applied to keep the text element consistently visible. Triggered by the delegation method called at the ending of text editing, the view slides back again all the way that was previously calculated.



Figure 6.9: Screenshot: possible positions of text elements related to the on-screen keyboard in Sketch It! for iPad

Text elements in Sketch It! for Mac are editable after a double-click

Sketch It! for Mac does not have to take care of a on-screen keyboard covering parts of the canvas since it is controlled by a physical keyboard. Instead, I had to find a solution to realize both the moving of text elements and their editing. Since they are realized as `NSTextField` objects recognizing mouse events, I was able to detect clicks on them. But in contrast to the behavior of `UITextView`s in the iPad implementation that allow editing after a completed touch gesture, `NSTextField` objects trigger editing of their content at mouse down.

To also enable moving of text elements, I adapted their property to be editable according to the count of mouse clicks. All text objects located on the canvas are not editable until the user performs a double-click on them — thereby, they are both movable and editable. Nevertheless, the moving of text elements starts with a short delay since the App awaits the time interval in which a second click might be executed. If the user performed a single mouse click in this interval, moving of the text element starts.

Chapter 7

Evaluation

“We all want progress, but if you’re on the wrong road, progress means doing an about-turn and walking back to the right road; in that case, the man who turns back soonest is the most progressive.”

—C. S. Lewis

To evaluate the Sketch It! system, a qualitative user study was conducted. I arranged a brainstorming session with the CEO and employees of a business company that focuses on the business areas software development and IT consulting. By investigating the application of the groupware to a real-world scenario, I was able to evaluate if the design requirements (see section 6.1—“Design Requirements”) for the software were met and how well the implementation of Nomadic Operations was accepted.

Sketch It! was evaluated in a brainstorming session with a relevant user group

7.1 Qualitative User-Study

The brainstorming session was carefully organized to meet the recommendations by Charles Clark (see section 2.4—“Brainstorming”). I observed the session together with Sören Busch, who assisted me with the conduct.

The brainstorming session was organized based on previous research

Four days before the meeting, participants were invited by the company's CEO with an email describing the brainstorming rules and formulating the topic for the session — in order to give the session participants an opportunity to already mull over the problem. The topic formulation (translated from German) read as follows:

"The company provides €30.000 to increase the motivation and productivity of its employees. In which purchases should be invested?"

A spade question was asked

According to Clark [1989], the proposed question is a spade question (see section 2.4—"Brainstorming"). It is formulated as clearly as possible and leaves no room for deviating comprehensions. The topic limits the budget to be spent and focuses on purchases instead of other expenses like activities. Hence, the ideas expressed by the group members were based on a common understanding.

The session took place on a Friday afternoon between 2:30 and 4:30 pm. The appointed time was suitable due to the facts that the brainstorming participants were in relaxed mood because of the upcoming weekend and the existence of a sufficiently large time interval after the lunch break to avoid fatigue.

The established brainstorming practices of the company have drawbacks

Usually, brainstorming sessions in the company's meetings follow one out of two different patterns. Both kinds are moderated by a person who is also responsible for note taking. In the first variant, the moderator keeps track of ideas in a digital text document, presentation file or project plan. The alternative is based on paper notes: the group members record their ideas on paper, which are merged after the meeting by one participant. It is obvious that the established workflows create well-known problems of brainstorming sessions (see section 2.5—"Limiting Factors for Brainstorming") such as production blocking, because the group members have to consider others' contributions and wait for the secretary to keep up.

7.1.1 Participants

The set of participants consisted of one woman and six men, who were aged between 19 and 41 with an average age of 29 (rounded). Their professions constituted a large variety: apart from the CEO, a secretary, the leader of the department for software development, a software developer as well as a student and two trainees took part in the meeting. The work experience of the group members in their current profession ranged between eight months and 14 years. Their rates of participation in brainstorming were diverse: while two participants frequently took part in brainstorming sessions (about once a week), others participated in brainstorming once a month or did not attend a brainstorming session before.

Seven participants with varying professions took part in the brainstorming session

Charles Clark [1989] recommends choosing a mixture of participants with varying backgrounds, attitudes, and specialties to increase productivity and generate a manifold set of ideas. In addition, the presence of both genders is an advantage for the motivation of group members. The set of participants for the prototype evaluation fulfills these recommendations. Besides, the group size of seven participants conforms well to the statements by the respondents of the survey described in section 4.1—“Survey: Collaborative Work in Business Meetings”: a large amount of business meetings consisting of seven participants was classified as satisfying.

The compilation of participants provides several advantages

7.1.2 Set-Up

The study took place in a dedicated meeting room well known to the participants. The environment was calm and brightly illuminated, and the room was furnished with a large table and chairs for each participant.

We provided six iPads running Sketch It! for iPad and a 40 inch-sized LCD display that was enhanced with a SMARTBoard to enable touch and pen-based user input. The display was connected to a MacBook Pro running Sketch It! for Mac. The Nomadic Server was also executed



Figure 7.1: Set-up for the Sketch It! user study. The equipment consisted of six iPads with pens and a MacBook Pro connected to a SMARTBoard.

by this computer. All devices established a wireless network connection to an access point installed in the meeting room. To enable pen-based interaction with the iPads, we provided styluses that the users could utilize as an alternative to sketching with their fingers. Besides, a Bluetooth keyboard could be used for typing in Sketch It! for Mac. A photo of the complete study set-up is shown in Figure 7.1.

Video recording kept track of the user interaction

The brainstorming session was captured on video. We installed a video camera connected to a laptop next to the SMARTBoard in order to keep track of the work performed at the table. Additionally, a screencast captured the user interaction on the large display.

At the beginning of the meeting, the users had to fill in a consent form in order to give permission to the anal-

ysis of their ideas and behavior. Afterwards, I gave a short introduction into brainstorming in general and presented the brainstorming rules described in section 2.4—“Brainstorming”. Subsequently, the topic was again formulated and displayed on an additional screen to be visible during the whole session.

After that, the Sketch It! system was demonstrated in detail. I presented its tools and range of functions while the participants had the opportunity to reproduce the interaction steps simultaneously. If the users had any queries regarding the groupware, they were encouraged to propose them. Furthermore, the control of Nomadic Operations was explained until everyone confirmed understanding of their handling.

The Sketch It!
system was
explained in-depth

The session consisted of two phases. The first phase was meant for idea generation and lasted 15 minutes. The participants were supposed to sketch their ideas on the iPads and the SMARTBoard. Communication and interexchange were allowed. Subsequently, ideas were presented in the second phase. Based on the insights from the paper-based brainstorming session described in section 4.2—“Paper-Based Brainstorming Session”, the presentation phase was scheduled much longer than the idea generation phase (30 minutes), because the previous user study pointed out that much time is necessary for this job. Because of the established practice to incorporate a moderator into the meeting, the CEO was supposed to guide the idea presentation.

The session
consisted of an idea
generation phase
and a presentation
period

Besides, the requirement for idea serialization mentioned in the evaluation of the paper-based brainstorming session was taken into account: the meeting participants were supposed to send their ideas to the large display and explain their thoughts. The group members were encouraged to record additional ideas whenever they came to their minds.

Serialization of idea
presentation was
applied

Sören and me attended the whole session without any interference. We took handwritten notes about our observations that were discussed with the participants in retrospective interviews after the session. The gained insights are described in the following.

7.1.3 Observations

Idea Generation Phase

At the beginning of idea generation, one participant started taking notes on the SMARTBoard, while others worked with the iPads. Except for one person who was watching around, the group members immediately started generating ideas. Several participants were smiling because of the unusual situation.

The users utilized diverse variant types for idea recording

Idea recording was performed in various fashions: some users started typing, others took handwritten notes, and certain people created sketches. Several of them were complex and occupied the whole canvas space. The user interacting with the SMARTBoard used keyboard shortcuts to create new sketches. He did not utilize the typing tool but took solely handwritten notes.

The participants were creative and concentrated

Overall, the participants were very productive and concentrated. There were some breaks in which group members watched around at other iPads and the SMARTBoard, which triggered generation of new ideas. Also the passing of a motorcycle on the street in front of the building reminded one participant of a new proposal. The session proceeded in a relaxed atmosphere — some participants joked around. There were few conversations, but some talking about possible software extensions was present. The CEO collected notes about the system on a separate sketch.

After expiration of the first phase, the group was reminded to proceed to idea presentation.

Idea Presentation Phase

Ideas were successively presented on the SMARTBoard

The idea presentation phase was moderated by the CEO standing in front of the SMARTBoard. He asked the other participants to utilize the Nomadic Operations in order to send their sketches to the public display. Specifically, he invited his employees to simultaneously send their ideas and processed them in a last in-first out fashion.

Each idea that was currently presented remained at full size on the large display. After idea explanation by its creator, the moderator minimized the sketch by pressing the corresponding button in the tool window. He grouped related ideas by moving them next to each other and piling them — a screenshot of this sorting is shown in Figure 7.2.

Related ideas were grouped in piles

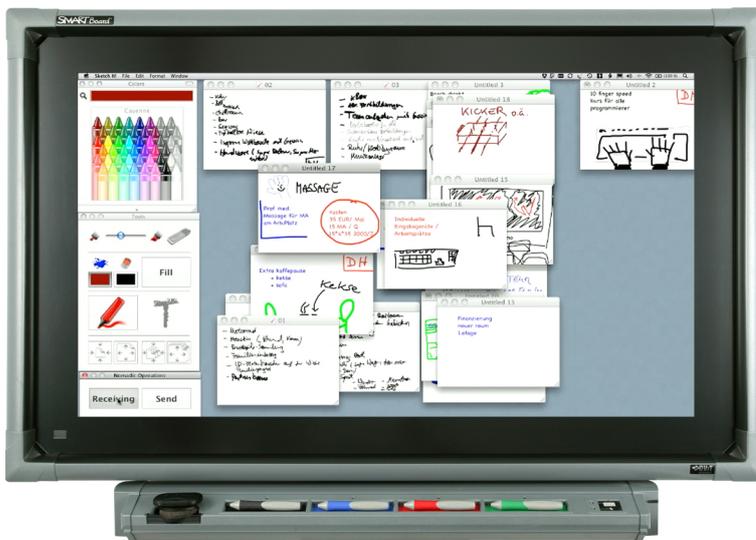


Figure 7.2: Screenshot: grouping of related sketches in piles

At one time, the “Minimize All” operation (see section 6.2.4—“Organization”) was selected, which inadvertently positioned the sketches in a grid layout ruining the grouping of sketches. At another point, the moving of a large sketch did not succeed, because the top bar of the document window was not hit. Instead, the user touched the canvas, which resulted in an added brush stroke. This action was reversed with the undo entry from the menu.

Most of the participants were interested in the discussion of presented ideas. Some group members generated additional ideas (which were often sent to the display as complementing ideas) and improved their previously created sketches. Overall, the accuracy of sketches varied among different participants. While some of them paid particular attention to the correctness of notes, others’ ideas contained several typos. During progress of the session, the moderator also created new sketches on the SMARTBoard

Ideas were still modified and extended

and added as well as corrected sent ideas. One idea was also dictated by a participant sitting at the table.

Users pursued different approaches to idea recording

Interestingly, the users pursued different approaches to record their ideas. Some participants wrote several ideas onto one sketch and drew connections or illustrations in a concept map-style. Others recorded one idea per sketch, as originally expected by us.

At the end of the time period scheduled for the presentation phase, the moderator limited speaking time for each participant to one minute, which caused some group members to talk faster. At this point, one user utilized his Pad Window to show an idea to the others by lifting the iPad.

7.1.4 Retrospective Interviews

In order to clarify my observations and propose a selection of follow-up questions, I conducted retrospective interviews with several group members.

Nomadic Operations were utilized for idea collection

One participant intensively used his iPad in the idea presentation phase. It became apparent that he activated the "Receiving" function in his App at each distribution of ideas. Thereby, he collected all other participants' ideas on his iPad. In the context of this behavior, the group came to the insight that serial contribution of ideas to the Board Window would have been more efficient than the simultaneous sending, which lead to an increased need for idea organization by the moderator.

The reason explaining the fact that some users recorded several ideas on one sketch was also mentioned: these participants roughly added notes to the sketch as they came to their minds. At this point, they did not know exactly how to express them and if they were related to previously created notes (that might allow for drawing connections between them). Additionally, they perceived the collection of notes on one sketch as timesaving. Group members who created one sketch per idea already knew exactly what to note and did not think further about it during their sketch creation.

Some participants wrote their initials into corners of the sketches. As a reason for this decision they mentioned that it was important for them to recognize their own sketches in the idea collection on the Board Window. Personalization features for ideas are useful to them. Other participants interposed that a default personalization of sketches (e.g., color coding or automatic include of names) might raise the problem of evaluation apprehension (see section 2.5—“Limiting Factors for Brainstorming”). Hence, the decision for labeling a sketch with identification details should be optional.

Personalization of sketches was differently evaluated

Participants praised the high flexibility of the software with respect to its features for idea recording. They mentioned having a “hands-on feeling” when creating their sketches and liked to design them exactly the way they intended to — as opposed to dictate ideas to the moderator who tries to retain them. Several group members liked the opportunity for taking immediate handwritten notes that could be corrected at a later point, because they felt to be as fast as possible and interacting more naturally when writing instead of using the text tool (which was in turn preferred by others who liked the visual appearance and possibility of repositioning). The participants stated that they could effectively use the software and believed to improve on their interaction with the system over a longer period when becoming more familiar with it.

People liked the various ways for idea recording

In addition, the users mentioned several ideas for improving the software. Some of them are listed below:

- Provide a copy feature for single sketch elements in order to migrate them from one sketch to another
- Include tagging functionality to semantically order several ideas
- Add a selection of cliparts and the function to paste own photos into the sketches
- Implement a set of several predefined geometric shapes (e.g., circle, square, etc.)
- Incorporate more colors (especially yellow) into the iPad App in order to extend visualization possibilities

- Change display of sketches in the iPad App to a grid layout instead of a list view and abandon the information about the last modification date
- Implement a countdown display in order to permanently visualize the remaining time for the session

7.1.5 Results

After the brainstorming session, the participants were provided with printouts of a post-study questionnaire in order to retrieve their evaluation of the brainstorming session and the Sketch It! system. The answers were given in a Likert scale with five equidistant points for each Likert item. A complete list of the proposed questions can be found in Appendix B—“Questionnaire for the Sketch It! User Study”.

The given answers are represented by numeric values

In the following, the given answers are summarized. I assign numeric values to the points of the Likert items and denote the most positive answers with the value five and the most negative answers with the value one. Hence, mean values below three classify negative evaluation and results greater than three are positive. The tables included below show the median, the mean value, and the standard deviation (SD) for the given answers to selected questions.

Participants are satisfied with the meeting

The overall satisfaction of the users, and ratings for motivation and the quality of their ideas are summarized in Table 7.1. In total, the evaluation is clearly positive: the mean value for the rating of the proposed brainstorming question amounts to 4.14 with a marginal standard deviation of 0.69. The ratings for personal motivation and quality of ideas range between 3.57 and 4.14, which constitutes a reserved but positive result. Therefore, we conclude that the meeting was adequately structured and organized. This result is additionally supported by answers to the question about the meeting environment: the mean value for appropriateness of the environment is 3.42. The time intervals for idea generation and presentation were rated at an average of 3.43 and 4.14. These values indicate that participants desired to have more time than available, especially for the presentation phase.

Question	Median	Mean	SD
How satisfied are you with the structure of the brainstorming session?	4	3.71	1.25
How interesting was the proposed question for you?	4	4.14	0.69
How motivated were you to participate in the brainstorming session?	4	4.14	1.07
How do you rate the quality of your ideas?	3	3.57	0.79
How much did you contribute to the brainstorming session?	4	3.57	0.98
How satisfied are you with your own performance in the brainstorming session?	4	3.71	0.49

Table 7.1: Satisfaction & motivation of the participants

The answers to major questions from the system evaluation are visualized in Figure 7.3. The heights of the bars are influenced by the amount of neutral answers and the distribution of positive and negative ratings: the more users decided for the neutral answer, the narrower the bars get. Their lengths towards the maximal and minimal values vary according to the intensity and count of positive and negative evaluations by the participants.

In consideration of the fact that the Sketch It! system is a prototype that builds the basis for additional improvements and extensions, the results constitute a success. The opinion of users on the software is significantly positive. Highest ratings were achieved for the system usability, which result in an average user satisfaction of 3.57.

The Sketch It!
system is rated
positive

We asked the users about their opinion on the devices that represented Tangible Windows in the brainstorming session. A summary of the given answers is shown in Table 7.2. Overall, the application of iPads and a SMARTBoard is perceived as useful. While the evaluation of iPads resulted in a moderately positive mean rating of 3.71 with a standard deviation of 1.50, the SMARTBoard received the highest ratings among all proposed questions. Except for one

The provided devices
are appropriate
support

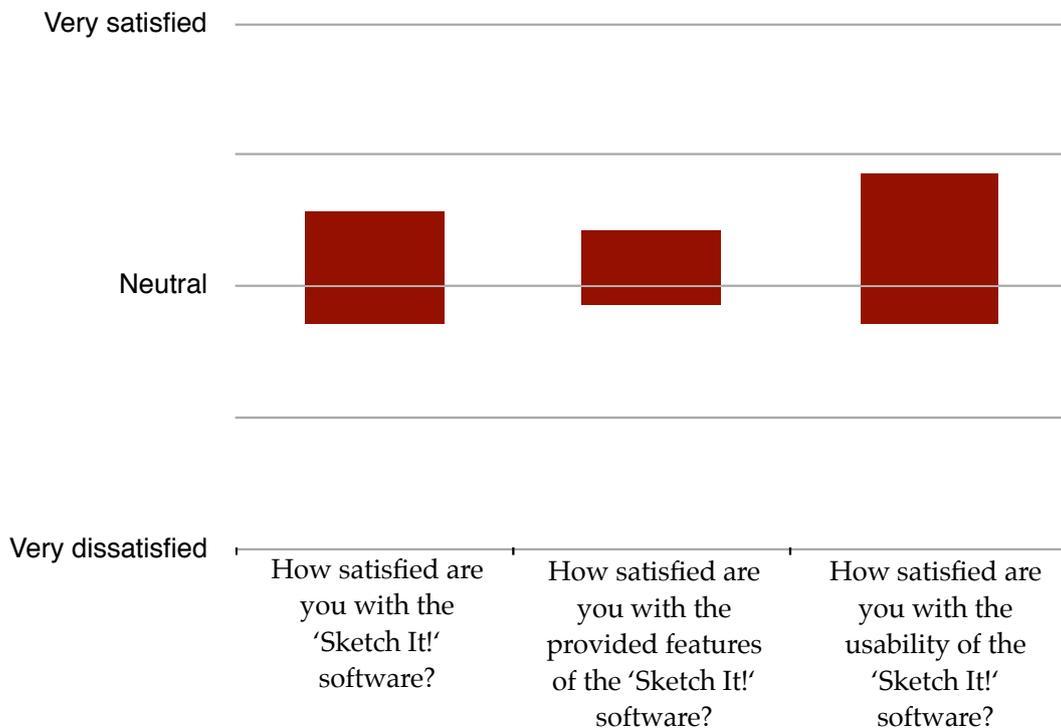


Figure 7.3: User satisfaction with the Sketch It! system

participant, all group members classify a SMARTBoard as "very useful", which results in a mean value of 4.86 and a median of 5.

Nomadic Operations are proven to be suitable

The evaluation of Nomadic Operations is a success: the users rated them as helpful for their collaboration. The mean value for their answers is 4.43 and the standard deviation 0.79, which is very low. This result is also reflected by the value 5 for the median. Based on this insight, we conclude that Nomadic Operations are suitable support for collaborating groups.

Brainstorming with Sketch It! is more productive than the previous procedure

In comparison to the established brainstorming procedure of the company, the Sketch It! system was classified as an improvement. We received one neutral answer — all other participants perceived the system as a more suitable support for their brainstorming sessions. The mean value for the given answers to the corresponding question is 4.00 with a standard deviation of 0.58. This insight is strength-

Question	Median	Mean	SD
How useful do you rate the application of iPads in a brainstorming session?	4	3.71	1.50
How useful do you rate the application of a SMARTBoard in a brainstorming session?	5	4.86	0.38
How do you rate the implementation of Nomadic Operations in the software?	5	4.43	0.79
How do you rate the application of the 'Sketch It!' system in comparison to your familiar brainstorming procedure?	4	4.00	0.58
Could you image to incorporate the 'Sketch It!' system in your future brainstorming sessions?	4	4.29	0.76

Table 7.2: Evaluation of equipment & Nomadic Operations

ened by the results for the question if users could imagine utilizing the groupware in their future brainstorming sessions: three participants “strongly agreed” to this question, three “agreed” to it, and one person was neutrally disposed. These results lead to the implication that the Sketch It! system is an appropriate tool for brainstorming.

Participants had the opportunity to explicitly formulate significant differences between their familiar brainstorming procedure and the interaction concept behind Sketch It!. The main differences lie in the idea visualization: the users liked the flexible text and sketch elements that could be extended and corrected at a later point, which is as improvement over paper-based idea recording. Users also mentioned to have more fun in brainstorming when using Sketch It!.

Flexibility and fun are major advantages of the Sketch It! system

Additionally, the Nomadic Operations for simple and effective idea exchange constitute an advantage in comparison to the familiar idea recording by a single secretary, who distributes the notes as digital documents via email.

The prototype development was limited to support idea generation, but not structuring	The respondents mentioned few ideas for improving the brainstorming process, which gives us confirmation of an appropriate organization and design for the user study.
Users suggested several improvements for existing tools	In addition to the improvement suggestions from the interviews (see section 7.2—“Summary”), we collected further proposals through the questionnaire. Several participants desired additional features for structuring and sorting of ideas, such as the creation of concept maps from generated sketches or filing them into semantic groups. The success of a brainstorming session also depends on the combination and filtering of information as well as prioritization of ideas after their generation, which is described in section 2.4—“Brainstorming”. Hence, functionality for supporting subsequent work after the brainstorming session is necessary and reasonable. Existing meeting software like ThinkTank (see section 3.3.2—“ThinkTank”) provides features for these tasks. However, the Sketch It! system is a vertical prototype and its implementation was completed at the level to support idea generation — assistance for idea structuring is a suitable extension to the system that should be addressed in future development.
The system evaluation confirms fulfillment of the design requirements	There are several ideas for enhancements of existing tools from the software: users would like to rotate text elements and change their color at a later point. Furthermore, they propose moving of brush strokes on the canvas and deleting them at once. Additionally, some participants desired the possibility to specify larger brush sizes and to change them in discrete steps in order to meet the exact value of a previously selected size when it was changed in the meantime. This is especially useful for people who take handwritten notes.

7.2 Summary

The system evaluation showed that Sketch It! is an appropriate support for collaborative brainstorming. Based on the answers from questionnaires, observations, and retrospective interviews, we can conclude that the design requirements for providing a simple user interface and var-

ious ways for idea recording (see section 6.1—“Design Requirements”) are fulfilled by the software. Furthermore, the several advantages of the groupware over previous brainstorming practices established in the company demonstrate that the design requirement to minimize common brainstorming hindrances is also met. Because of the users’ praise of correction and modification features, we confirm that error recovery tools such as undo and redo functionality are successfully implemented into the software.

We wanted to investigate if Nomadic Operations are appropriate support for a collaborative workflow with Tangible Windows. Users rated them as very helpful and utilized these features extensively during the brainstorming session for different purposes. One user made use of the “Receiving” operation in order to collect others’ ideas on his Pad Window. Every group member utilized the “Send” operation to demonstrate ideas to the others.

Nomadic Operations and Tangible Windows improve collaboration

Besides, the device classes to which the design of Sketch It! was oriented were classified as appropriate. For instance, user interaction during the meeting exhibited a situation in which one participant lifted the Pad Window for demonstration purposes. This observation shows flexibility of the iPad devices. In combination with the positive results of the evaluation of the SMARTBoard, we verified the suitability of the provided device classes.

The device classes are suitable

The users mentioned various additional application scenarios, in which the Sketch It! system might also provide support for collaborating users. One could imagine utilizing the software for interactive trainings: the participants have to work on exercises that are subsequently sent to a person who corrects them and includes annotations. Afterwards, the results are redistributed among the group members or get presented to the whole group for demonstration purposes. Also use cases in classes at school or university might be realistic: teachers and lecturers would be able to easily provide data to their audience. Every receiving person could take the information with her or extend it by own ideas.

There are several application scenarios for Sketch It!

It becomes obvious that the Tangible Windows concept and

collaborative work go well together. The users who evaluated my groupware in a practically relevant brainstorming session perceived several advantages of the system and are disposed to incorporate Sketch It! into future meetings.

Chapter 8

Summary and Future Work

*“There is no other solution to man’s progress
but the day’s honest work, the day’s honest
decisions, the day’s generous utterances and the
day’s good deed.”*

—Clare Boothe Luce

The Tangible Windows concept constitutes a new approach to realize natural and flexible interaction with data, and exchange of information. The concept takes the windows from the screen of modern desktop computers and brings them into the physical world. By utilizing a combination of hardware components in various form factors, it lets users interact with data independent from a single device. Tangible Windows provide Nomadic Operations, which enable the exchange of information between them.

This last chapter summarizes the work on my project and the results derived from the system evaluation. In the second part of the chapter, I will give suggestions about further development of the prototype and future research on the topic.

8.1 Summary and Contributions

In this thesis, I presented the design and development of the groupware “Sketch It!”. The system is based on the Tangible Windows concept and applies this approach to a collaborative scenario: brainstorming. The final implementation is a fully functional software prototype, which can be utilized to accomplish real-world tasks for idea generation and note taking.

Characteristics of collaborative work have been characterized

Meetings are an established part in the working schedules of today’s business companies. These situations constitute both opportunities and challenges in providing computer-based support. At the beginning of my work, I studied literature investigating collaborative work and issues that can be addressed by computer systems in order to support interacting groups with their tasks.

Analysis of meetings gave an insight into collaborative workflows

Afterwards, we conducted a survey with people who frequently participate in business meetings. The analysis of their experiences lead to the insight that brainstorming is a common technique for collaborative generation of ideas and that the success of meetings is closely related to their organization. Following this, we set up a brainstorming session based on recording of ideas with pen and paper. In this session, we observed how people interact in such situations and gained insights, which helped us to adequately design this kind of meetings. The results showed that a presentation phase for ideas including serial focusing on contributions by each participant is important for the productivity of the group. Additionally, we learned about different ways of idea recording: while some group members took textual notes, several meeting participants created sketches in order to illustrate their ideas.

Evaluation of storyboards confirmed plausibility of use cases

Based on these insights, I created storyboards as first prototypes, which illustrate possible use cases for groupware that supports user input as texts and sketches. In order to address tasks occurring in real-world settings, I formulated user stories in combination to the storyboard illustrations, which emphasized possible drawbacks in meeting situations and presented our vision how to address arising issues with Tangible Windows in combination with Nomadic

Operations. The storyboards were evaluated by potential users who are experienced with meeting situations. We achieved results confirming that the depicted situations are plausible use cases for our groupware.

Following a set of design requirements that was derived from our previous results, I designed and developed the groupware Sketch It!, which was implemented for iOS and Mac OS X. Its design and set of features have been iteratively refined. The final iteration in the development of the prototype resulted in a fully functional system. It is a sketching application with additional support for text editing. Additionally, we implemented a framework into the groupware, which enables flexible exchange of data: the Nomadic Operations Framework.

To evaluate Sketch It!, we conducted a user study in which employees from a software company participated in a brainstorming session. It has been organized based on previous research on brainstorming and insights from the user study about paper-based brainstorming. The system evaluation showed that Sketch It! is an appropriate support for collaborative brainstorming. Users effectively incorporated the system into their meeting and expressed demand to utilize the groupware in their future brainstorming sessions.

The contribution of this work is a confirmation that the Tangible Windows concept provides reasonable support for collaborating groups. The users are satisfied with the available tools and perceive them as improvement for their group work. Based on the results from our system evaluation, we conclude that the design requirements for the system have been fulfilled. Additionally, we proved that Nomadic Operations provide a benefit for interacting groups. From these results, we infer that we attained the research goals formulated in section 1.0.1—“Research Goals”.

For future systems based on the Tangible Windows concept, it needs to be assured that the users are provided with a selection of features for information sharing and content exchange: we received positive feedback from users who liked the flexibility of the system generated by the incorporation of several devices (Tangible Windows) and simple data exchange (by utilizing Nomadic Operations). There-

Sketch It! supports texts and drawings in combination with functionality for data exchange

The system evaluation could show the usability and effectiveness of Sketch It!

My research contribution is a demonstration of appropriateness for the collaborative utilization of Tangible Windows and Nomadic Operations

Emphasizing flexible usage and simple data exchange are important design guidelines for future systems

fore, we believe that the Tangible Windows approach is justified for varying workflows.

8.2 Future Work

The Sketch It! system was perceived as suitable support for brainstorming sessions by the users evaluating the groupware. They mentioned different improvement suggestions and the study rose additional research questions, which appear reasonable to be investigated in the future.

8.2.1 Software Improvements

Idea Organization

As described in the summary of results from our user study (see section 7.2—“Summary”), the Sketch it! implementations are vertical prototypes limited at a level that does not provide support for subsequent work on generated ideas. Organization tasks like filtering, ordering, and prioritizing cannot be accomplished with the software.

Implementation of tools for subsequent work after the brainstorming sessions is important

These tasks are important for the quality of the outcomes of a meeting. Experts need to prioritize notes and distinguish the meaningful suggestions from inappropriate proposals. Therefore, an implementation of tools providing functionality for tagging of sketches, grouping them into concept maps (or folder structures similar to file systems) as well as delegating personalized ideas as task formulations to group members will increase the acceptance of the system.

Further study of existing tools is necessary

In addition, existing features for idea organization such as the automatic layout of sketches need to be further investigated. At some points during our evaluation, the moderator had to search for the appropriate button in the GUI to activate minimization of all sketches. There is obviously potential to simplify the user interface for these operations and make the widgets easier differentiable for the user.

Additional System Features

Users suggested several improvements for the Sketch It! system that might be implemented in future versions of the software. For instance, additional colors in the iPad version would be useful. One could imagine designing a custom widget in order to provide the user with a selection from the whole color spectrum (similar to the color widget in Mac OS X).

Comfortable color selection from the complete spectrum is suitable

An increased set of functionalities for the existing tools would also improve the software. Additional modifications for text and path elements as well as further export formats (i.e., images) for sketches might be suitable. Users also mentioned ideas for different display of existing sketches: one solution would be including a grid view into the iPad App, which the user could order and modify according to her needs.

Additional export features and more powerful tools would be useful

The quality of the sketches could be enhanced by applying smoothing algorithms to the Bézier Curves on the canvas. Thereby, the visual appearance of the brush strokes would be increased and the readability of handwritten notes could be improved. Besides, it needs to be examined if such beautification makes sense: while the visual appearance of ideas is increased, the character of being preliminary vanishes — which might not be desired by some user groups.

Smoothing of paths would increase sketch quality and readability of handwritten notes

Additional User Studies

The system evaluation was performed with potential future users of the software in a realistic work scenario. The user group consisted of seven participants, which is a suitable group size for brainstorming.

However, the system needs to be utilized by additional groups in order to increase the amount of user feedback and to investigate how users with varying expertise and habits utilize the groupware in different environments.

Additional evaluation scenarios will increase the amount of feedback

A long-term user study would generate a confirmation of durable acceptance

Besides, it needs to be investigated how well the system is accepted by meeting participants if it is steadily available to them. If they perceive it as support for their collaboration, they would frequently utilize the groupware. The confirmation of such long-term acceptance is still pending.

The software could be freely distributed through the App Stores

A possible approach to evaluate the system with a larger user group would be the distribution of the software via [iTunes](http://itunes.apple.com/)¹ and the [Mac App Store](http://www.apple.com/mac/app-store/)² by Apple. If we made the two Sketch It! variants available as free downloads, a group of volunteering users could incorporate the system into their meetings and afterwards provide feedback.

8.2.2 Additional Application Scenarios

Sketch It! should be evaluated in different kinds of meetings

The survey about meetings in business companies described in section 4.1—“Survey: Collaborative Work in Business Meetings” revealed a diverse set of topics addressed in today’s meetings. Apart from brainstorming, there are tasks like evaluation processes and project planning. In order to orient the Sketch It! system to a broader set of application scenarios, it needs to be evaluated when utilized in meetings that are dealing with these tasks. Additionally, one has to investigate which system features might be implemented to support other meeting scenarios.

Remote collaboration constitutes a special application area

In its current state, the system prototype is based on wireless network connections that are established by the clients. However, remote collaboration is also a reasonable application scenario.

If the groupware would support information exchange from devices that are moved out of a meeting environment (e.g., by users who have to leave but would still like to participate), group members could contribute ideas from other locations. This may be technically realized by data connections established through mobile radio.

¹<http://itunes.apple.com/genre/mobile-software-applications/>

² <http://www.apple.com/mac/app-store/>

The proposals by users from the system evaluation are further suggestions that should be taken into account: it needs to be examined how well the Tangible Windows concept and the Sketch It! system fits the needs of collaborating groups in education — for instance in school or university. The application to medical or technical business fields, such as software development (e.g., to support the creation of UML diagrams) is also imaginable.

Additional application fields need to be investigated

My research focused on collaborative tasks and investigated the suitability of the Tangible Windows concept and Nomadic Operations for group work. It is furthermore important to examine the workflows of single users and their needs at personal workspaces. From these results, one could derive design requirements for TW-based Apps to support single users. We believe that outcomes from both fields will reasonably complement each other.

The Tangible Windows concept and Nomadic Operations need to be applied to single-user scenarios

8.2.3 Nomadic Operations

At the moment, Nomadic Operations are realized through a client-server architecture (see section 6.3.1—“Nomadic Operations Framework”). In order to increase flexibility of the system and to develop the software independent from the server, the framework needs to be revised to a peer-to-peer architecture. The devices representing Tangible Windows could then establish a connection between each other instead of using a central point.

The Nomadic Operations Framework needs to be revised to a peer-to-peer architecture

Another useful extension could be implementing an add-on to existing applications the users are familiar with. If the framework would be able to extend the functionality of these Apps by Nomadic Operations, we could extrapolate the set of application scenarios.

Implementing Nomadic Operations into existing Apps could be an advantage

Furthermore, we have to study which additional Nomadic Operations might be useful. Feedback from the Sketch It! user study contained a suggestion providing synchronizing functionality between sender and receiver in order to keep the data consistent on both Tangible Windows. Examining further possible operations seems to be an adequate research field.

An investigation of additional Nomadic Operations is promising

Location-awareness
could improve
Nomadic Operations

It is also worth investigating how to identify the various devices to each other and how to execute the Nomadic Operations: one possible solution for the future could be introducing location-awareness. As an example, this functionality would enable meeting participants to perform a swipe gesture in order to send data from one Tangible Window to another. The user could select the target window by directing her gesture to the location of another meeting participant sitting at the same table. Because of the location-awareness of the devices, the regarded group member would receive the sent data in her window.

8.2.4 Additional Form Factors for Tangible Windows

Devices in other form
factors might provide
suitable support

The development of the Sketch It! prototype was directed to two specific device classes: Pad Windows and Board Windows. We believe that the interaction concept can be extended to devices in other form factors as well.

Laptop computers are established devices in several business fields (see section 4.1.2—“Tools”). It appears reasonable to investigate incorporating these computers into the collaborative workflows with Sketch It!. Since users often bring laptops to meetings, the set of available tools providing possible support for the meeting participants would be extended.

As described in section 1.1.1—“The Vision of Ubiquitous Computing”, Marc Weiser [1999] mentions the device class *tabs* acting as digital post-it notes. One could imagine to use smart phones such as the [Apple iPhone](http://www.apple.com/iphone/)³ as Tab Windows providing users with the functionality to collect several minimized application windows as icons and migrate them from one environment to another. The content of the windows could then be transferred to additional devices suitable for the corresponding task.

Weiser explains the user interaction with tabs, which also seems worth investigating in the context of Tangible Win-

³<http://www.apple.com/iphone/>

dows, as follows:

“It will also act as an extension to computer screens: instead of shrinking a program window down to a small icon on the screen, for example, a user will be able to shrink the window onto a tab display. This will leave the screen free for information and also let people arrange their computer-based projects in the area around their terminals, much as they now arrange paper-based projects in piles on desks and tables.”

[Weiser, 1999, p.5]

Appendix A

Survey Questions: Collaborative Work in Business Meetings

The survey questions shown on the following pages were proposed to the participants on a website. In order to integrate them in a convenient way, I converted the HTML form into text and tables.

COLLABORATIVE WORK IN BUSINESS MEETINGS - 1 / 5

This survey is supposed to collect information about the collaborative work in meeting situations.

When you participate in a meeting, how do you exchange information, which tools and devices do you incorporate and where do you experience advantages and drawbacks in you collaborative work?

Your contribution will serve as support for the creation of a software prototype providing tools for collaborative work on Apple iPads. The information contributed through this survey is anonymous and will be treated confidentially.

If there are questions you are not able to answer in English: German contributions are also fine.

** Required*

PERSONAL INFORMATION

1. **What is your gender? ***
 - Male
 - Female
2. **What is your age? ***
3. **What is the highest level of education you completed? ***
 - High School (Abitur, Realschule, Hauptschule)
 - Bachelor / Diploma (FH)
 - Master / Diploma
 - Doctorate
 - Other
4. **If you are a student or have completed your studies:
What is/was your major?**
5. **What is your main occupation? ***
 - Employee / Freelancer

COLLABORATIVE WORK IN BUSINESS MEETINGS - 2 / 5

- Self-Employed
 - Student
 - Pupil
 - I am currently not employed
 - Other
6. **If you are employed, what is the exact job description?**
 7. **In what kind of business or organization do you work?**
 8. **For how long are you holding your current main occupation? ***

THE MEETING SITUATION

The following questions collect information about the frequency and structure of your meeting appointments.

9. **What was your average participation rate for meetings in the last year? ***
 - Less than once a month
 - About once a month
 - Two or three times a month
 - About once a week
 - Almost every day
 - More than once a day
10. **In how many COLLABORATIVE MEETINGS do you participate? ***

"Collaborative Meeting" means productive work with an outcome generated by several people located in the same room.

 - none
 - few
 - many
 - most
 - all

COLLABORATIVE WORK IN BUSINESS MEETINGS - 3 / 5

11. What are the topics of your meetings and which is the most popular collaboration method for each topic? *

"Collaboration method" denotes the meeting style, i.e. brainstorming, evaluation, discussion, ...

12. How are your collaborative meetings structured? *

Please describe shortly how your collaboration is usually coordinated. Is there a moderator? Is the agenda posted? Ahead of time? Who is taking notes? Are they distributed? How? ...

13. How many of your overall meeting appointments consist of the following group sizes? *

	none	few	many	most	all
2 Participants					
3 Participants					
4 Participants					
5 Participants					
6 Participants					
7 Participants					
8 Participants					
9 Participants					
10 Participants					
More than 10 Participants					

14. Given the group size, how many collaborative meetings are on average satisfying for you? *

	none	few	many	most	all
2 Participants					

COLLABORATIVE WORK IN BUSINESS MEETINGS - 4 / 5

3 Participants					
4 Participants					
5 Participants					
6 Participants					
7 Participants					
8 Participants					
9 Participants					
10 Participants					
More than 10 Participants					

PLEASE ANSWER THE FOLLOWING QUESTIONS WITH YOUR MOST FREQUENT MEETING IN MIND

15. What are factors that prevent your meetings from being successful? *

16. Which tools are used in your meetings? Are they appropriate for keeping record of your work? *

	NOT USED	not helpful	less helpful	helpful	very helpful
Pen & Paper					
Flip Chart					
Laptop					
Tablet Computer					
Desktop/Media Computer					
Digital Projector					
Smartphone					

COLLABORATIVE WORK IN BUSINESS MEETINGS - 5 / 5

Blackboard					
Whiteboard					
Digital Whiteboard					
Wall Screen					
Interactive Wall Screen					

- 17. Are there other tools you incorporate into your collaborative work?
Which tools? Are they helpful?**
- 18. Why are the tools you rated "less helpful" or "not helpful"
inappropriate for your meeting support?**
- 19. How could your collaboration be improved?**

Appendix B

Questionnaire for the Sketch It! User Study

The questionnaire shown on the following pages was presented to the participants as a printout. I translated it from German before including it into the thesis.

Questionnaire for the Sketch It! User Study

The analysis of the following questions will be performed anonymously

OVERALL SATISFACTION

1.) How satisfied are you with the structure of the brainstorming session?

--	--	--	--	--

Very satisfied

Very dissatisfied

2.) How interesting was the proposed question for you?

--	--	--	--	--

Very interesting

Very uninteresting

MOTIVATION & PERFORMANCE

3.) How motivated were you to participate in the brainstorming session?

--	--	--	--	--

Very motivated

Very unmotivated

4.) How do you rate the quality of your ideas?

--	--	--	--	--

Very good

Very bad

5.) How much did you contribute to the brainstorming session?

--	--	--	--	--

Very much

Very little

6.) How satisfied are you with your own performance in the brainstorming session?

--	--	--	--	--

Very satisfied

Very dissatisfied

7.) Do you rate the environment of the brainstorming session appropriate?

--	--	--	--	--

Very appropriate

Very inappropriate

AVAILABLE TIME

8.) How do you rate the length of the brainstorming session?

--	--	--	--	--

Too short

Exactly right

Too long

9.) Did you have enough time during the idea generation phase?

--	--	--	--	--

Too short

Exactly right

Too long

10.) Did you have enough time during the idea presentation phase?

--	--	--	--	--

Too short

Exactly right

Too long

SYSTEM EVALUATION

11.) How satisfied are you with the 'Sketch It!' software?

--	--	--	--	--

Very satisfied

Very dissatisfied

12.) How satisfied are you with the provided features of the 'Sketch It!' software?

--	--	--	--	--

Very satisfied

Very dissatisfied

13.) How satisfied are you with the usability of the 'Sketch It!' software?

--	--	--	--	--

Very satisfied

Very dissatisfied

14.) How useful do you rate the application of iPads in a brainstorming session?

--	--	--	--	--

Very useful

Very useless

15.) How useful do you rate the application of a SMARTBoard in a brainstorming session?

--	--	--	--	--

Very useful

Very useless

16.) How do you rate the implementation of Nomadic Operations in the software?

--	--	--	--	--

Very useful

Very useless

17.) How do you rate the application of the 'Sketch It!' system in comparison to your familiar brainstorming procedure?

--	--	--	--	--

Much better

Equivalent

Much worse

18.) Could you imagine to incorporate the 'Sketch It!' system into your future brainstorming sessions?

--	--	--	--	--

General agreement

Strong disapproval

19.) What are the main differences between the 'Sketch It!' system and your familiar brainstorming procedure?

--

20.) Which improvement suggestions do you propose for the execution of the brainstorming process?

--

21.) Which improvement suggestions do you propose for the 'Sketch It!' system?

--

22.) In which additional application areas could the 'Sketch It!' system be used?

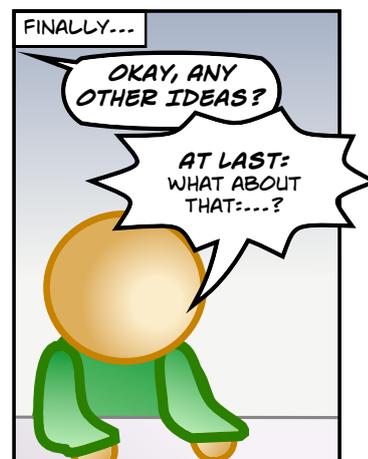
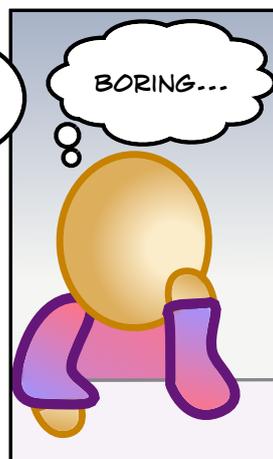
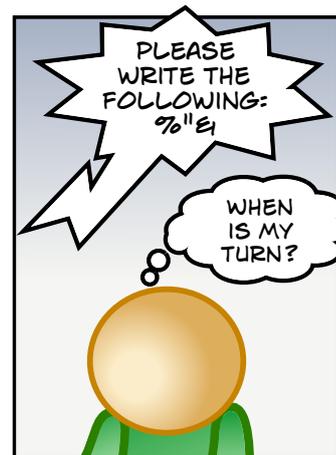
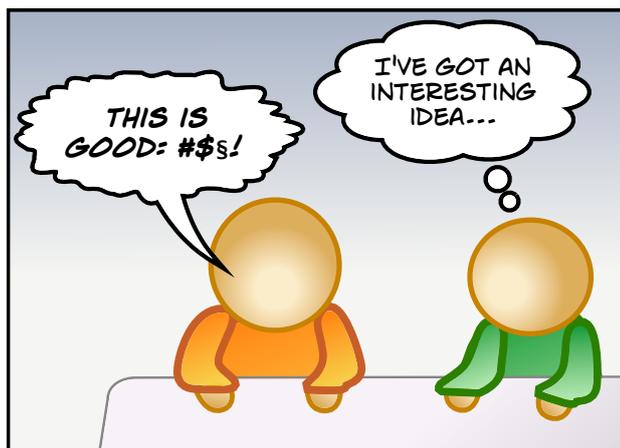
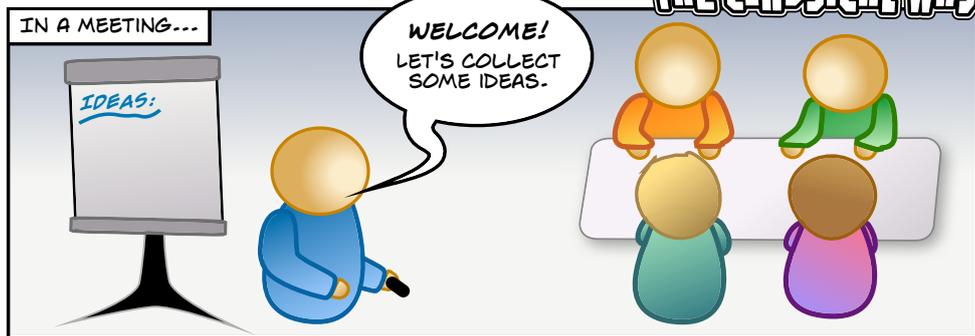
--

Appendix C

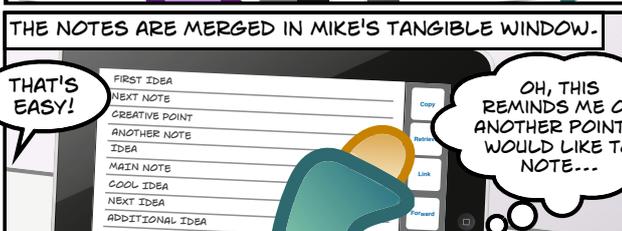
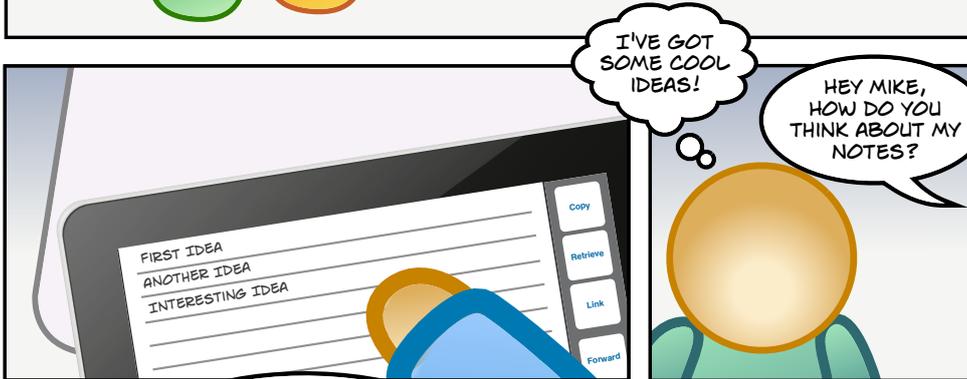
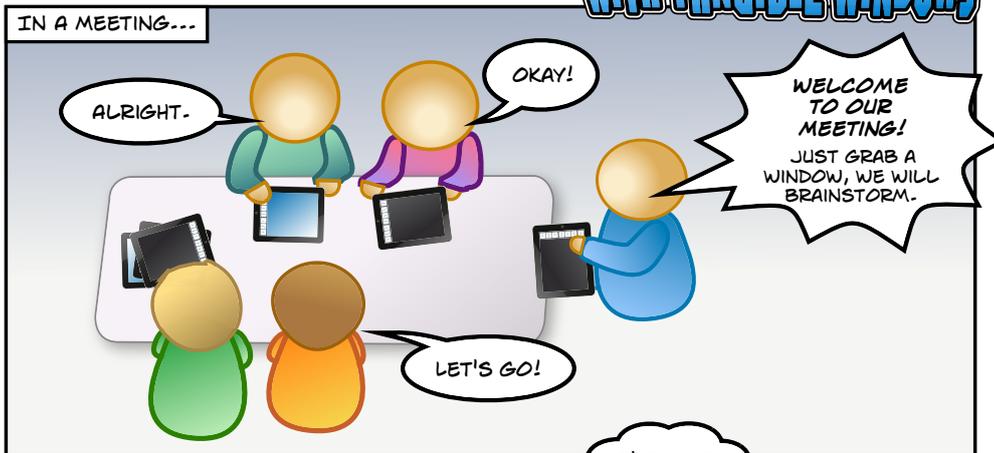
Storyboards

COLLABORATIVE BRAINSTORMING

THE CLASSICAL WAY

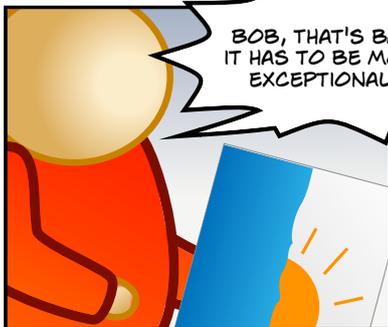
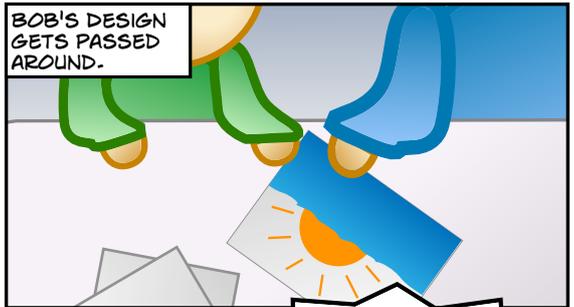
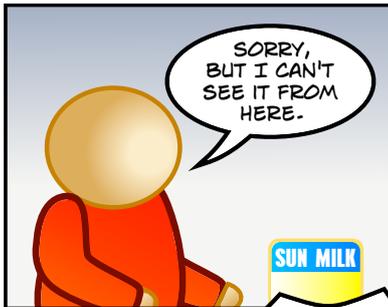
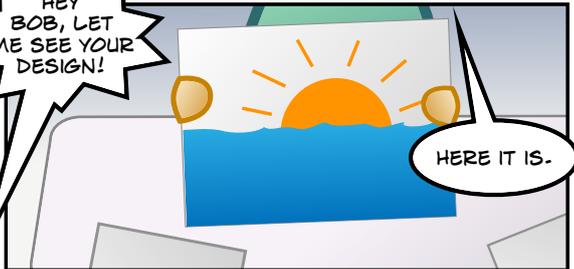
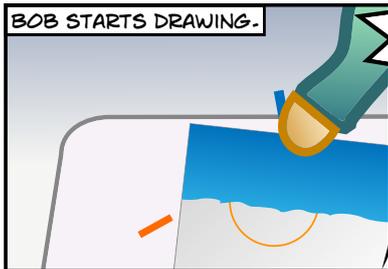
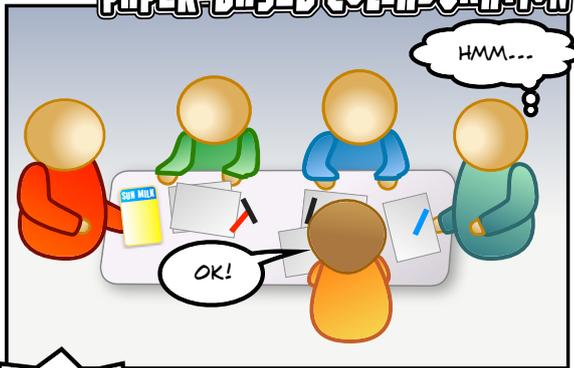
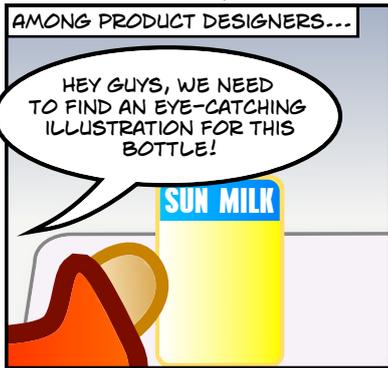


COLLABORATIVE BRAINSTORMING WITH TANGIBLE WINDOWS

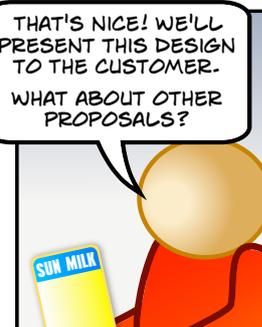
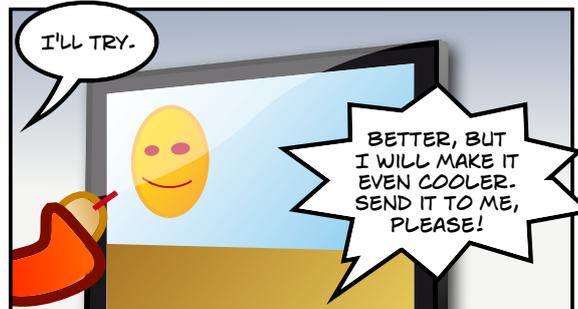
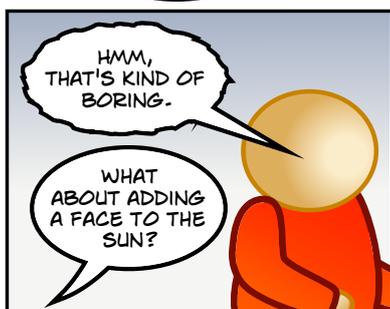
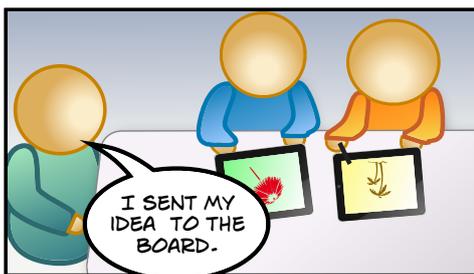
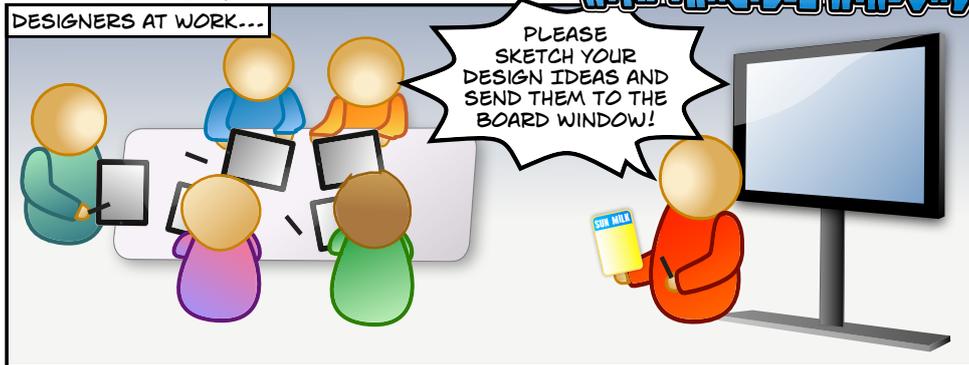


DESIGN SESSION

PAPER-BASED COLLABORATION



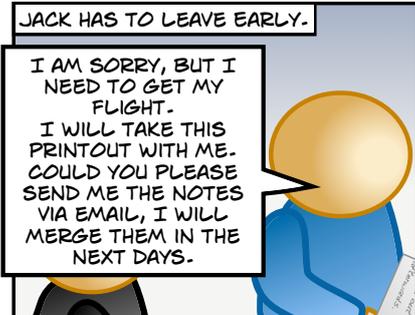
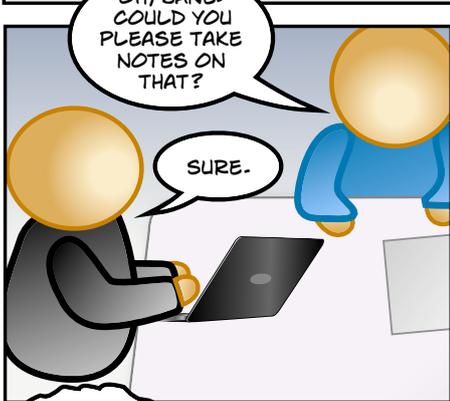
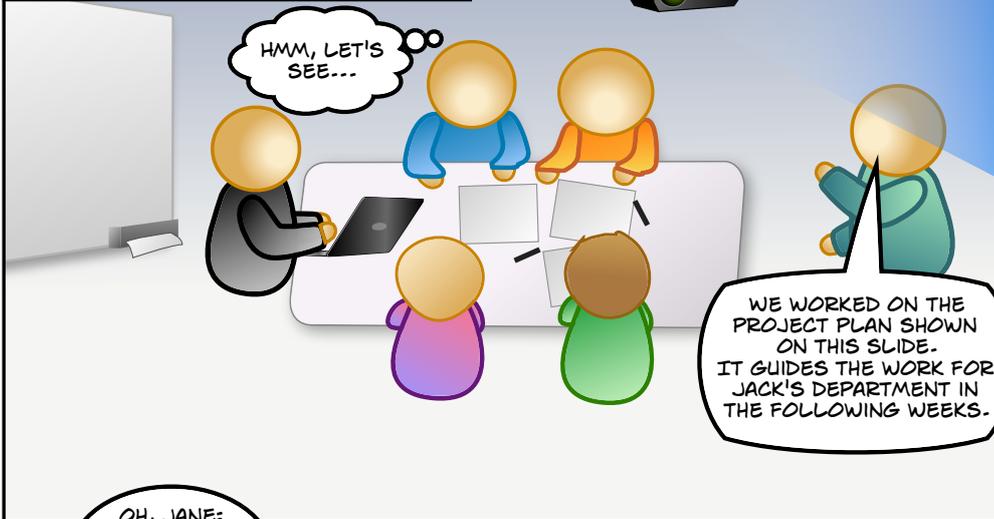
DESIGN SESSION WITH TANGIBLE WINDOWS



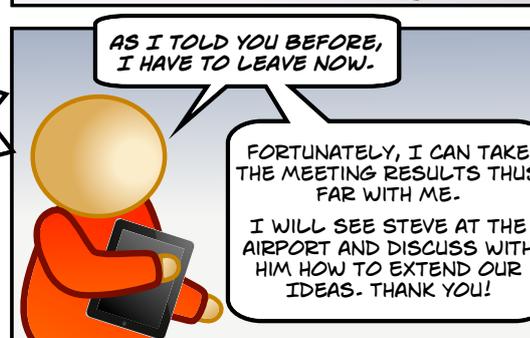
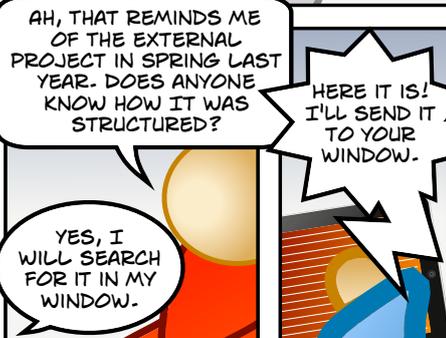
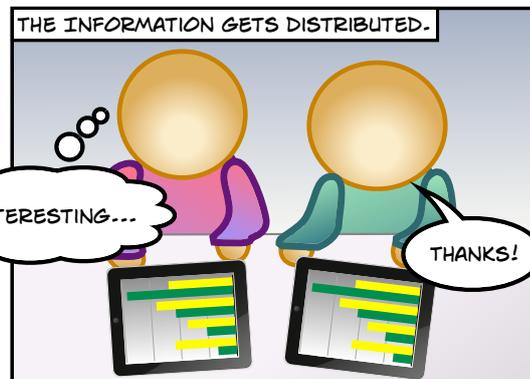
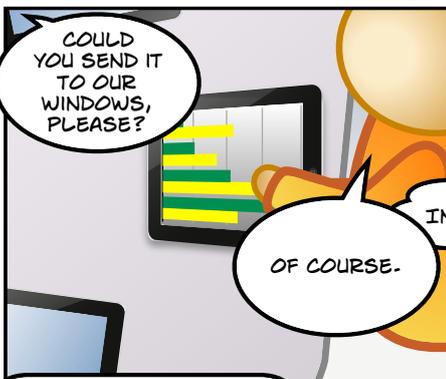
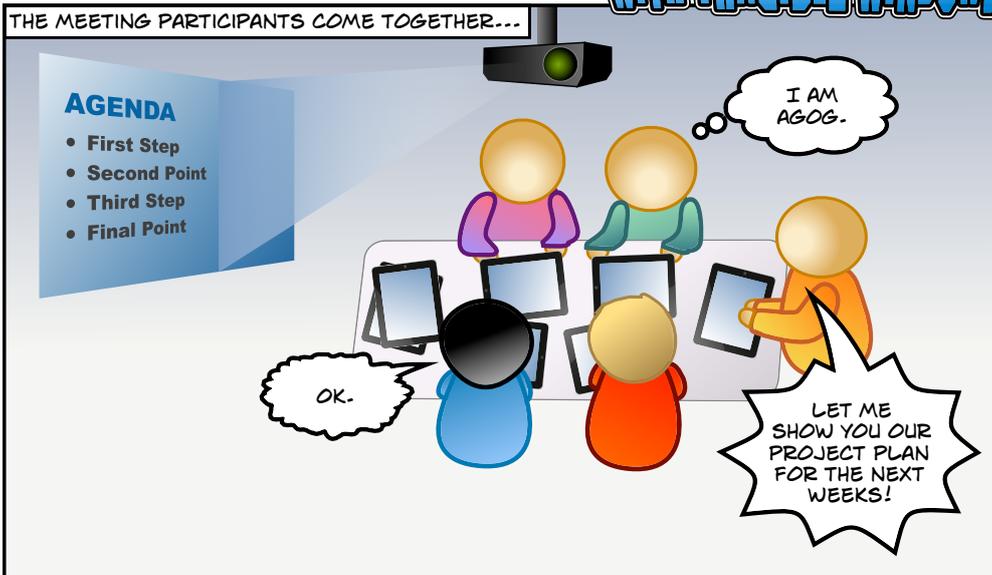
STRATEGIC MEETING

A MEDIUM MIXTURE

IN A BUSINESS MEETING, DEPARTMENTS PRESENT THEIR PROJECT CONTRIBUTIONS.



STRATEGIC MEETING WITH TANGIBLE WINDOWS



Bibliography

James H. Bair. Supporting cooperative work with computers: Addressing meeting mania. In *COMPCON Spring '89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers.*, pages 208–217, 1989.

Liam J. Bannon and Kjeld Schmidt. *CSCW: Four Characters in Search of a Context*, pages 3–16. North-Holland Publishing Co., Amsterdam, The Netherlands, 1991. ISBN 044488811X.

Jacob T. Biehl and Brian P. Bailey. Aris: an interface for application relocation in an interactive space. In *Proceedings of Graphics Interface 2004, GI '04*, pages 107–116, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society. ISBN 1568812272.

Jacob T. Biehl, William T. Baker, Brian P. Bailey, Desney S. Tan, Kori M. Inkpen, and Mary Czerwinski. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, CHI '08*, pages 939–948, New York, NY, USA, 2008. ACM. ISBN 9781605580111.

Charles H. Clark. *Brainstorming: How to Create Successful Ideas*. Wilshire Book Company, 1989. ISBN 0879804238.

Alan R. Dennis, Joseph S. Valacich, and Jay F. Nunamaker Jr. An experimental investigation of the effects of group size in an electronic meeting environment. *IEEE Transactions on Systems, Man and Cybernetics*, 20(5):1049–1057, 1990. ISSN 0018-9472.

Michael Diehl and Wolfgang Stroebe. Productivity loss in brainstorming groups: Toward the solution of a riddle. *Journal of personality and social psychology*, 53(3):497–509, 1987.

Joan Morris DiMicco, Anna Pandolfo, and Walter Bender. Influencing group participation with a shared display. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04*, pages 614–623, New York, USA, 2004. ACM. ISBN 1581138105.

Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction*. Prentice Hall, 3. edition, 2004. ISBN 0130461091.

Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. Groupware: Some issues and experiences. *Communications of the ACM*, 34:39–58, January 1991. ISSN 0001-0782.

Dr. Floyd J. Fowler Jr. *Improving Survey Questions: Design and Evaluation*. Sage Publications, 1995. ISBN 9780803945838.

Irene Greif, editor. *Computer-Supported Cooperative Work: A Book of Readings*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 093461575.

GroupSystems. *Quick Start Guide to ThinkTank 3*, April 2011. URL <http://www.groupsystems.com/pdf/QuickStartManual.pdf>.

Joshua Hailpern, Erik Hinterbichler, Caryn Leppert, Damon Cook, and Brian P. Bailey. Team storm: demonstrating an interaction model for working with multiple ideas during creative group work. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition, C&C '07*, pages 193–202, New York, NY, USA, 2007. ACM. ISBN 9781595937124.

Michael Haller, Jakob Leitner, Thomas Seifried, James R. Wallace, Stacey D. Scott, Christoph Richter, Peter Brandl, Adam Gokcezade, and Seth Hunter. The nice discussion room: Integrating paper and digital media to support collocated group meetings. In *Proceedings of the 28th international conf, CHI '10*, pages 609–618, New York, NY, USA, 2010. ACM. ISBN 9781605589299.

- Charles M. Hymes and Gary M. Olson. Unblocking brainstorming through the use of a simple group editor. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work, CSCW '92*, pages 99–106, New York, USA, 1992. ACM. ISBN 0897915429.
- Brad Johanson, Armando Fox, and Terry Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1: 67–74, April 2002. ISSN 1536-1268.
- Saadi Lahlou, editor. *Designing User Friendly Augmented Work Environments*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 1848000979.
- Jacob Nielsen. Iterative user-interface design. *Computer*, 26 (11):32–41, November 1993. ISSN 0018-9162.
- Donald A. Norman. *The Design of Everyday Things*. Basic Books, New York, reprint paperback edition, 2002. ISBN 0465067107.
- Alex F. Osborn. *Applied Imagination: Principles and Procedures of Creative Thinking*. Charles Scribner's Sons, 1953. ISBN 9780023895203.
- Jef Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 0201379376.
- Abigail J. Sellen and Richard H.R. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003. ISBN 9780262194648.
- Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-land: an interactive landscape for creativity and innovation. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, CHI '99*, pages 120–127, New York, NY, USA, 1999. ACM. ISBN 0201485591.
- Wolfgang Stroebe and Bruno S. Frey. Self-interest and collective action - the economics and psychology of public goods. *British Journal of Social Psychology*, 21(Jun):121–137, 1982.

John C. Tang. Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34(2):143–160, 1991. ISSN 0020-7373. Special Issue: Computer-Supported Cooperative Work and Groupware. Part 1.

Jennifer Thom-Santelli, Dan Cosley, and Geri Gay. What do you know? experts, novices and territoriality in collaborative systems. In *Proceedings of the 28th international conference on Human factors in computing systems, CHI '10*, pages 1685–1694, New York, USA, 2010. ACM. ISBN 9781605589299.

Mark Weiser. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3: 3–11, July 1999. ISSN 1559-1662.

Kipling D. Williams, Stephen G. Harkins, and Bibb Latané. Identifiability as a deterrent to social loafing: Two cheering experiments. *Journal of Personality and Social Psychology*, 40:303–311, 1981.

Index

- access point 100
- Active Space *see* Augmented Work Environment
- affordances 3
- API 88
 - Cocoa 88, 90
 - Cocoa Touch 88, 90
- App 75
- App Store 118
- ARIS 33
- Augmented Meeting Room. *see* Augmented Work Environment
- Augmented Work Environment 26, 51

- Bézier Curve 79, 88, 92
- Bonjour 89
- brainstorming 17–21, 64–65, 97, 114
 - idea evaluation 21
 - idea recording 20
 - invitation 20, 98
 - participants 19
 - room 20, 99
 - rules 18, 55, 98
 - spade question 19, 98
 - steam-shovel question 18
- brainstorming hindrances 21, 50, 75
 - evaluation apprehension 22, 24, 75
 - free riding 22, 24
 - production blocking 21, 23, 75, 98
- brainstorming, paper-based 54–59

- collaboration 13, 15–17, 44
- communication 12, 16
- Computer-Supported Cooperative Work 11, 25
- convex hull 79
- coordinate system 93
- CSCW *see* Computer-Supported Cooperative Work

- DIA Cycle 7

- evaluation 7, 71, 97

-
- font 92
 - form factor 4, 120–121

 - groupware 14, 25, 54, 73
 - non-real-time 14
 - real-time 14
 - GUI 74

 - i-Land 26
 - IMPROMPTU 34
 - iOS 75, 115
 - iPad 2, 61, 73, 102, 111, 117
 - iPhone 120
 - iRoom 29
 - iterative design 7–9, 15, 81, 115
 - iTunes 118

 - Likert scale 106
 - location-awareness 4, 27, 120

 - Mac 2, 73
 - Mac OS X 2, 76, 115, 117
 - MacBook Pro 99
 - mean 106
 - median 106
 - meeting 1, 43
 - meeting software 36
 - middleware 32
 - mode error 16, 34, 82

 - NiCE Discussion Room, The 30
 - Nomadic Operations 6–7, 54, 61, 72, 75, 87, 101, 109, 119–120
 - framework 89, 119
 - Receiving 7, 87, 111
 - Send 7, 34, 87, 111

 - Objective-C 88
 - on-screen keyboard 80, 95–96
 - Open Source 89

 - prototype 2, 73
 - low-fidelity 8, 63
 - vertical 110, 116

 - Sketch It! 73–97, 107
 - brush/text color 78, 117
 - class model 90–93
 - design requirements 74–75
 - drawing canvas 78
 - eraser 80, 81
 - fill color 79

-
- sketching 79–80
 - typing 80–81
 - SMARTBoard 76, 87, 99, 102
 - smoothing algorithm 117
 - standard deviation 106–108
 - storyboard 63–65, 114

 - tabletop 1
 - Tangible Windows 2–6, 64–69, 74, 111, 115
 - TEAM STORM 37
 - ThinkTank 39, 110
 - TW *see* Tangible Windows

 - Ubiquitous Computing 4–6, 26, 120
 - boards 4
 - pads 4
 - tabs 4, 120
 - user story 64–69, 114
 - user study 55–61, 97–110, 115, 117–118

 - Virtual Network Computing 33
 - VNC *see* Virtual Network Computing

 - website 70
 - widget 78

