NiCE Brainstorming: An Interactive Environment to Support the Combination of Verbal and Electronic Brainstorming

Florian Perteneder

DIPLOMARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Juni 2011

Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Hagenberg, am 24. Juni 2011

Florian Perteneder

Contents

Acknowledgment

Before diving into the exciting topics of brainstorming and interaction design, I would like to thank my family for supporting me in whatever I'm doing. I'd also like to thank my supervisor Michael Haller for his excitement in the topic and his continuous support. Developing NiCE Brainstorming and writing this Thesis would not have been possible without the help of my colleagues from the Media Interaction Lab, especially Thomas Seifried, who is a great help when dealing with software architecture, Jakob Leitner, who always has a new idea in terms of interaction and screen design and Christian Rendl, who was available to ask a short question at all times. Last but not least I would also like to thank my fellow students, who helped me with taking photos or in other ways.

Kurzfassung

Obwohl Studien seit geraumer Zeit darauf hinweisen, dass elektronisches Brainstorming mehr Ideen produziert, wird die traditionelle, mündliche Form doch viel häufiger angewendet, um strukturiert Ideen zu entwickeln. Jedoch ist die reine Anzahl der Ideen möglicherweise nicht das einzige aussagekräftige Maß zur Bestimmung der Produktivität. Einige Forschungsergebnisse deuten darauf hin, dass die niedrigere Produktivität bei der klassischen Form des Brainstormings durch andere Vorzüge aufgewogen wird. Um die Vorteile von mündlichem und elektronischem Brainstorming zu verbinden, wurde mit NiCE Brainstorming eine interaktive Umgebung entwickelt, die nicht nur beide Methoden unterstützt, sondern auch einen nahtlosen Übergang zwischen beiden Techniken ermöglicht. Ein interaktives Whiteboard mit einer speziell entwickelten Software ermöglicht traditionelles, mündliches Brainstorming in einer digitalen Umgebung; ferner können verschiedene Geräte wie Computer und Smartphones, aber auch digitales Papier verwendet werden, um Ideen festzuhalten und dem Brainstormingprozess in asynchroner und verteilter Form beizusteuern. Durch ein neues und einfaches Bedienkonzept ermöglicht die Anwendung auf dem interaktive Whiteboard eine schnelle und flüssige Interaktion. Die Einbindung des Web-basierten Dienstes *Evernote* zum Speichern von Notizen und Ideen ermöglicht zusätzlich die Unterstützung einer große Anzahl von elektronischen Geräten. Auch wenn das System bis jetzt nur informell getestet wurde, haben dabei vor allem die Möglichkeiten zur Strukturierung von Inhalten auf dem interaktiven Whiteboard einen sehr guten Eindruck hinterlassen. Da die derzeitige Umsetzung sicher auch noch einige Verbesserungsmöglichkeiten zulässt, verspricht gerade die Kombination von traditionellem und elektronischem Brainstorming großes Potenzial.

Abstract

Though research indicates that electronic brainstorming provides higher productivity, verbal brainstorming is still the most popular method to create ideas in a structured way. The number of generated ideas is most probably not the only meaningful measurement, since the collaborative process of face-to-face brainstorming provides several advantages that outbalance the lower productivity. In order to combine the advantages of verbal and electronic brainstorming, NiCE Brainstorming, an interactive environment was created, not only to enable both methods in a single system but also to support fluid switching between the two ways of brainstorming. It includes an interactive whiteboard application to digitize the process of verbal brainstorming. Moreover, various devices, including computers and Smartphones but also digital paper can be used to contribute ideas in an asynchronous and distributed way. While the whiteboard application provides fluid and fast interaction by introducing a novel and slim interaction concept, the number of supported external devices was increased by using the web-based note taking service *Evernote*. Although the system was only evaluated informally, we believe in the great potential of the approach, especially regarding the powerful structuring capabilities on the whiteboard. However, we also identified several issues to improve the environment that supports the freedom in the process of brainstorming even further.

Chapter 1

Introduction

Innovation and creativity are catchphrases in modern economy. Companies and people have to be imaginative to be successful. Numerous of success stories tell how global companies, like Google[1] or Facebook[2], are built on top of one ingenious idea. However, how do people come up with their ideas? When it comes to the systematical search for new ideas, the term *brainstorming* comes up to most people's minds. Since the nineteen-sixties, numerous creative techniques have been developed. Brainwriting, Brainwriting pool, 6-3-5, or Collective Notebook [32] are just the tip of the iceberg, besides initially invented brainstorming. These and other methods appear in various management guidebooks and training courses.

1.1 Brainstorming in Use

Though most people come in touch with brainstorming during education or even use it within their daily work-life, what they call brainstorming is often different with what Osborn [24] proposed. In many cases, the term brainstorming is simply used for a meeting when the goal is to generate new ideas and come up with new features, products, or solutions. This does not mean an idea generation process is of less quality, when it does not follow the proposed rules, because often there are reasons why the search for ideas is performed in a special way. Since creating ideas is a highly personal issue and ideas are often generated by using associations, personal experience, or preferences, many people have adopted the process to their needs and use it the way it works best for them. Moreover, the task of its use is an extremely important issue. It is nearly impossible to use the same practice to search for a product's name also to develop a new company strategy. Anyway, last but not least many people also miss the knowledge about how to perform the method in its original or optimal way. It should be emphasized that

[1] http://www.google.com/

[2] http://www.facebook.com/

the original brainstorming method proposed by Osborn is not at all the best method, but it had a strong influence on the practical use of creative techniques and idea generation. Despite various research results indicate that people produce a higher number of ideas when they work separated [10, 25, 26], the dominant type of brainstorming remains a group activity based on face-to-face communication.

1.1.1 Definition

However, due to these various perceptions, it is necessary to sharpen the definition before writing about brainstorming. Actually the term brainstorming only includes the technique developed by Osborn, but for an expedient use within this Thesis, a more extensive definition is required, as it is implicitly used in most recent brainstorming literature [7, 26]. Brainstorming does not describe any process of idea generation, but the use of a specific technique - whatever it may be - to do so. It is considered as a group activity, where target-oriented methods are used to generate as many ideas as possible on a determined issue. The term group activity does not necessarily include any collocation or communication, but it states that all persons within the group are working toward the same goal and are going to combine their efforts at the end. Therefore, when people think about ideas on their own, it is not considered as brainstorming, as long as their individual work is not pooled at the end and, therefore, is explicitly part of a superior process.

1.1.2 Process of Brainstorming

Generating ideas do not solve any problem. At first it is necessary to frame the problem. In a second step, actual ideas are created and at the end the most promising idea is selected and hopefully realized. Since the development of the original brainstorming technique, many improvements were presented that tried to adapt the original process or to develop other creative techniques, in order to increase productivity[3]. However, the term productivity only refers to the second step of the whole process. High productivity does not help to frame the problem or to select the best idea. Various studies [7, 26] display the advantage of brainstorming techniques, as nominal and electronic brainstorming that use individual idea generation, but in companies, the production of many ideas is not everything [34]. Framing the right problem and selecting the best idea is also very important. If a group consists of experts from different fields, it is possible to address an issue from many different perspectives. Among others, this may be a reason why verbal brainstorming is still extremely popular, since the exchange about problems but also about thoughts and visions is essential, and communication is disburdened by face-to-face brainstorming.

[3] Usually productivity is defined as the number of non-redundant ideas per group [26].

1.2 The Idea of NiCE Brainstorming

As stated earlier, many studies [10, 25] proofed the superiority of nominal and electronic brainstorming over face-to-face or rather verbal brainstorming. Despite these facts, many people, organizations, and companies still use verbal brainstorming, although it apparently is an inferior technique. Beside the assumption, that it is used to a great extent because it just is the most famous technique, modern research states various reasons for this circumstance. Verbal brainstorming provides group interaction, supports group well being, and offers a welcome contrast to ordinary work [7]. Moreover, thinking of brainstorming in a complex field, it is necessary to have experts around that are able to help by providing specialized knowledge when it is required [34]. But there are also evidences that the popularity of verbal brainstorming is mostly about distorted perception of productivity within a group [22, 34].

1.2.1 General Outline

These findings lead to the idea of creating a system that combines the advantages of verbal and nominal respectively electronic brainstorming. In doing so, the use of an interactive environment provides the necessary infrastructure to digitize the analog process of verbal brainstorming. This way the successful combination of the two processes is enabled. However, to build a fully functional system, which is not only capable of managing digitized verbal brainstorming sessions but also includes multiple external devices to support and to enhance characteristics of electronic brainstorming, plenty of issues have to be solved. For further investigation and implementation, the topic is divided into two separate issues, in order to build one multi-functional brainstorming environment. One goal is to support the concept of traditional, verbal brainstorming within a digitized and interactive environment. The other objective is to equip various analog and electronic devices with the ability to use them to contribute ideas.

1.2.2 Face-to-face Brainstorming Environment

In order to support verbal brainstorming, it is necessary to arrange an environment that enables brainstorming of a collocated group in a digitized way. A look at a brainstorming session in a traditional environment identifies the used tools and utilities. Depending on the actual conduction of the session, most of the time a large public whiteboard (or blackboard) is used; sometimes a pin board is used in combination with small pieces of paper (e.g. Post-its[4]). Another alternative is a large sheet of paper, mounted on the wall, to record notes in a more permanent way than on a whiteboard. It is also helpful if no whiteboard is available or if more space is required. These

[4]Post-it is a registered trademark of 3M.

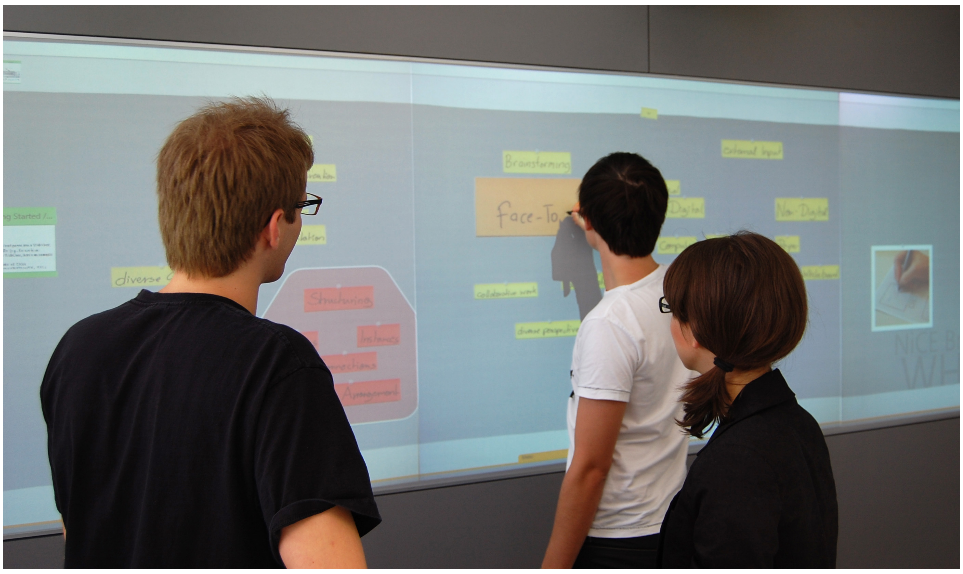
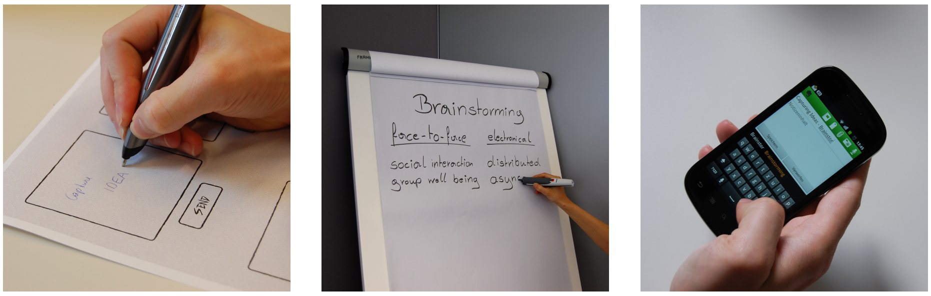


Figure 1.1: Using an interactive whiteboard to perform face-to-face brainstorming in a digitized environment.

observations show that space to display ideas is most essential. Moreover, it is necessary that the ideas are visible to all the participants taking part in the brainstorming session. Under these circumstances, the use of a large display to provide a digital environment for brainstorming seems an adequate solution. An interactive whiteboard fulfills most of the requirements, even though the use of a tabletop-based system or the combination of tabletop and interactive whiteboard would have been a possible solution. However, there is no issue of orientation on an interactive whiteboard, what results in a higher conspicuity of ideas. In addition, the combination of displays is not as widespread as single interactive whiteboards. Moreover, the use of a single display is more economic and easier to handle in terms of maintainability. Therefore, the digital brainstorming environment consists of a single, large-scale interactive whiteboard (see figure 1.1). It is essential that the software on the interactive whiteboard is simple and easy to use. It is not necessary to provide many features in the first place, but to develop a system that supports a fast and fluid workflow. Furthermore, the use of brainstorming software on an interactive whiteboard should provide additional value beyond storing capabilities, compared to a traditional non-digital solution.

1.2.3 Distributed Brainstorming Environment

The second goal is about the integration of external devices into the system. Since nominal or electronic brainstorming is considered as highly effective in terms of productivity, it is reasonable to add these capabilities to the sys-



(a) (b) (c)

Figure 1.2: Various external devices that can be used to contribute ideas. (a) Use of digital paper. (b) Creating ideas by using a digital flipchart. (c) Using Evernote on a Smartphone for submitting ideas.

tem. People should be able to choose what method fits best for their task and their team. Even combinations of the two types should be possible and encouraged. A good choice (from the scientific point of view) is to start off with a distributed session, where people think about the problem individually. Later on people join in a group setting, in front of the interactive whiteboard. In this situation, more ideas can be generated by building on top of other ideas. At the end, the evaluation and selection process is started. The main idea is to support many external devices (see figure 1.2), to allow idea contribution in various situations. Overall, there are two different categories of devices. Computers, notebooks, Smartphones or tablets are capable of communicating by using web-based interfaces or specific client software. More important, these devices are able to display external content by themselves. Therefore, they provide a way to access the ideas of others, what gives the user the possibility to build on top of these ideas. The other category includes mediums like paper, flip charts or whiteboards that are initially not capable of contributing ideas digitally. However, the use of tracking technologies enables digital contribution. Despite various calls for a paperless office, pen and paper are still around, so it seems reasonable to support them too.

Another variation between the two presented categories of devices is the difference in the capability of providing feedback. While it is no challenge to offer an interface that enables simple idea contribution on a Smartphone, this process is a lot more difficult when only a piece of paper and a digital pen are available. It may not be a challenge in technical terms but regarding the design of the interaction process, due to missing feedback capabilities.

1.2.4 Solution

These considerations lead to the implementation of a prototypical system called NiCE Brainstorming that was developed to exhibit the whole process.

Due to extensive requirements some areas were neglected, but the prototype is capable of executing all the intended functionality, except the reverse channel that enables external devices to access the public idea-pool.

The main focus was on the interface and interaction design of the whiteboard application that more or less represents the core of the whole system. As mentioned in section 1.2.2, the goal was to develop a concept that enables fast and fluid interaction. In order to do so, the idea was to forgo a menu or dedicated tools to minimize the overhead when interacting. Without these utilities, the integration of different features into a multi-user system was a challenge. The solution contained many principles of object-oriented user interface design that proved to be a powerful concept when fast, simple, and fluid interaction is more important than including lots of different features.

By integrating the web-service Evernote[5], numerous of external devices become useable. Alongside various devices that are able to use a specific Evernote client, every browser-equipped device can be used to contribute ideas by using the web interface. Separately, a custom client application was developed to enable the contribution of content by using digital paper. This way, a great variety of devices and mediums that are typically used to take notes or to record ideas can also be used to contribute ideas to a common digital idea-pool. The contributed content is immediately available within the whiteboard application to inspire others to build own ideas on top of it. Moreover, in the final phase of brainstorming, when searching for a solution, the content from all the different devices can be structured and looked through on the interactive whiteboard, to provide an overview and reference for a discussion.

[5] Evernote is a registered trademark of Evernote Corporation and used under license.

Chapter 2

Related Work

Even though, the combination of verbal and electronic brainstorming in the introduced, iterative way is a new idea in terms of including it into a single system, there are several projects that worked on a similar issues. The design of NiCE Brainstorming was most influenced by PostBrainstorm [14], but the argumentation about enhancements on the process of brainstorming is similar to Hilliges et al.[16] in many respects. Beside those two publications, there are some other papers and projects that focus on different aspects and provide different solutions for novel brainstorming.

2.1 PostBrainstorm

PostBrainstorm [14] is a brainstorming application catered to the brainstorming sessions of the design firm IDEO[1]. It provides a high-resolution information wall to present different types of digital content and supports the use of fluid, pen-based interaction.

In companies, there is often a dedicated room for a project, where the relevant material is cluttered on walls and tables. However, physical walls have certain limitations. Moreover, there is no easy way to store or search the generated data, and digital material needs to be printed to be available. It is also time-consuming to get physical information back into the digital world.

Since PostBrainstorm (see figure 2.1) was designed to create a fully digital environment that is capable to substitute traditional mediums within a brainstorming process, there were certain specifications to meet. First of all, it was necessary to provide high resolution to display multiple documents and tables next to each other. This need was fulfilled using a tiled rear-projection display with 64 dpi resolution, called *Interactive Mural* [40]. Other design goals were to preserve a clean screen without any desktop-like widgets and fluid interaction, because continuous interruptions like dialogues

[1] http://www.ideo.com/

Figure 2.1: Post Brainstorm enables the user to work with high resolution images, application windows and hand drawn material. All interaction is performed using a digital pen that can also be used to sketch notes or take annotations. (Courtesy of François Guimbretière et al.)

are not adequate to a whiteboard-like scenario, especially within multi-user collaboration. The application takes heavy use of the *FlowMenu* [15] that is invoked anywhere on the surface by pressing the meta-button on the pen. For instance, it enables to move or to select sheets, and there are many more commands to call. Content can be added by using a ceiling mounted camera and tangible crop devices, a local database, a VNC[2]-connection to other computers in the local network, or by writing directly on the wall. These connected computers can be controlled via pen, and a snapshot tool enables to crop stimulating material. Moreover, it is possible to push content on the wall by using a particular software on external devices like laptops or PDAs. The issue of limited space on the whiteboard is addressed with a technique named *ZoomScapes*, a location-based scaling mechanism on top of the surface. Whenever any object is dragged to the top of the interactive area, it is scaled down to reduce space consumption. This way, the upper zone of the whiteboard can be used to store information for later use.

While there were no efforts made to provide a high-resolution display, fluid interaction was one of the main design goals when developing NiCE Brainstorming. To enable even faster and easier interaction, making all operations accessable without a menu, was an additional goal. Moreover, the use of external laptops and PDAs to contribute ideas inspired the effort to integrate external devices, as well. However, while PostBrainstorm was designed to fit the specific requirements of IDEO, NiCE Brainstorming attempts to provide a flexible and easy to use environment.

[2]Virtual Network Computing

2.2 Designing for Collaborative Creative Problem Solving

Hilliges et al.[16] presents an *Electronic Brainstorming System* (EBS), embedded in a socio-technical environment. The goal was to create an easy-to-use brainstorming tool that combines the advantages of electronic and face-to-face brainstorming.

Although many studies indicate that electronic brainstorming produces more ideas and provides further advantages of digital data processing, the original face-to-face brainstorming is still the dominant type in professional life. This leads to the assumption that some advantages of face-to-face brainstorming cannot be measured by productivity measurements. Following this argumentation, it is a logical step to combine the two types.

The system was designed, based on the brainwriting technique [12]. Users are able to create a Post-it by drawing a square on the interactive table. After that they can express their idea on it. After finishing, the Post-it is shrunken down and becomes movable. This way, multiple persons can express their ideas at once. The main goal is to minimize the cost of interaction and communication to support social interaction. The system consists of an interactive table and a large wall display (see figure 2.2). These two displays are used for the different phases within the process. While the table is used in the initial generative phase, the wall display is used in the structural phase. Every time a new idea is created on the table, it appears on the wall display, as well. Evaluation showed that people like the wall display to gain an overview over all the generated ideas, even during generative phase. On the wall, further tools, like clusters that can be merged, moved

Figure 2.2: While table is used to write down ideas, the wall-display provides an overview over the existing ideas and enables structuring (Courtesy of Otmar Hilliges et. al).

and connected, are available. The performed study pointed out that quantity of idea production was equal to the corresponding paper-based process but other than when using the paper-based equivalent, the user benefits from data processing and storing capabilities of a digital system.

The process of writing down ideas and other available tools show similarities to NiCE Brainstorming, but there are unmistakable differences in the setup, since NiCE Brainstorming only uses a single interactive wall display. While the system of Hilliges et al. uses a dedicated gesture to create new content that is fixed in size, NiCE Brainstorming creates Post-its flexible in size, when users start to write anywhere on the interactive whiteboard. Moreover, NiCE Brainstorming provides more options regarding the visualization on the whiteboard and capabilities to visualize external content. However, the considerations why to develop such a system are very similar. Both argumentations agree that there is a lack in measuring the true values of face-to-face brainstorming. This leads to the paradoxical situation that the inferior technique (face-to-face brainstorming) is more dominant in professional life than the actual superior technique (electronic brainstorming). To benefit from the possibilities of digital data processing it is necessary to include the benefits of face-to-face brainstorming into the system. This can be accomplished by combining these two types of brainstorming within an interactive environment. Anyway, the big difference is the interpretation of this combination. Hilliges et al. actually digitizes the process of face-to-face brainstorming and this way makes it electronic. On the contrary, NiCE Brainstorming attempts to combine the two processes. In order to provide a seamless change between the techniques, the process of face-to-face brainstorming is digitized, because there is no way to perform electronic brainstorming within a non-digital environment.

2.3 The Designers' Outpost

The Designers' Outpost [17] is a tangible user interface, designed to support collaborative website design. Especially in the early design stage of website development, traditional whiteboards or other available large spaces, as walls and tables are used for developing, explaining and communicating ideas. Within this process, designers use conventional mediums like pen and paper to visualize their thoughts. A common practice is to write the content, the website is supposed to contain, onto Post-its that can be arranged in categories on the wall. This technique is called *affinity diagramming* [3]. However, there are drawbacks to this paper-centric process. Annotations and links between the Post-its often get out of sync. When notes are shifted around, there is no method of versioning (apart from photographing the wall), and there is no way to integrate remote participants.

The goal of Designers' Outpost (see figure 2.3) is to combine the natu-

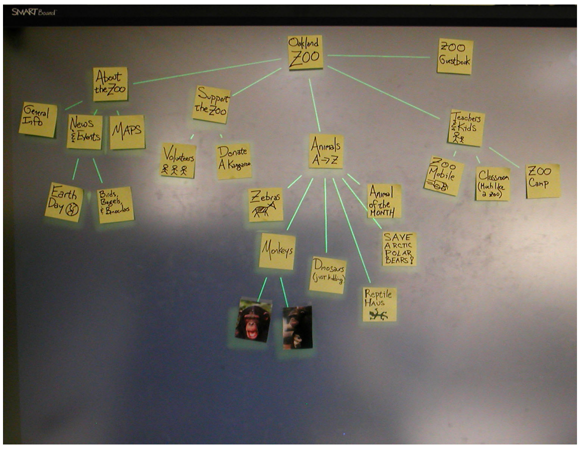


Figure 2.3: A web site information architecture visualized by Designers' Outpost using physical Post-its and images and virtual connections. (Courtesy of Scott Klemmer et al.)

ral interaction of the paper-centric approach with the power of interactive whiteboards to overcome these shortcomings. This is accomplished with a touch-sensitive whiteboard and a computer vision approach based on a two camera setup. A low resolution rear-mounted camera recognizes when Post-it tangibles are added or removed and triggers a high resolution front-mounted camera that captures the content. This enables replacing the physical Post-its by digital representations that can be moved and erased using specific tangibles. Moreover, pens can be used to connect items or to add content digitally by writing or sketching directly on the board.

Since NiCE Brainstorming does not support the use of tangibles, there is no need for a complex camera setup and computer vision recognition. Although the whole interaction is conducted digitally, we believe that this provides more flexibility in terms of the setup, because the application works with any interactive whiteboard technology that supports pens. To solve the stated problem of versioning, NiCE Brainstorming supports the concept of Instances that can also be used to create a snapshot of the current state.

2.4 Shared Sketches in Brainstorming Meetings

Within their paper *Encouraging Contribution to Shared Sketches in Brainstorming Meetings* [2], Bastéa-Forte and Yen present a collaborative sketching system that uses a shared canvas on individual tablet PCs and an interactive whiteboard. In traditional, verbal brainstorming sessions, usually one person acts as the moderator of the group. One of his tasks is to write down the ideas of the group. This can lead to bias according the recorded ideas. Moreover, it is often difficult to express ideas only verbally. Sketching is a widely spread technique, but there is a fairly high inhibition threshold for a person to stand up and explain an idea visually on the whiteboard. Furthermore, space is limited in front of a whiteboard, so people interfere with each other when they are sketching simultaneously.

A shared sketching system that mirrors the whiteboard to personal devices, such as tablet PCs, provides a way to sketch in a collaborative way and to give each member the possibility to contribute sketches without moving toward the whiteboard. In addition, people enjoy the possibility to develop their ideas without attracting too much focus. The result is often presented to the group after finishing. Whithin the evaluation, the developers found out that the use of the system lead to a more equal participation of the group members, but against prediction, the total amount of sketching per session decreased while the quality of the contributions increased. Another notable aspect of the shared sketching system was the reduction of face-to-face communication during the brainstorming process. Most of the verbal communication was substituted by visual communication, provided by the shared sketching canvas. Beside these changes in social behavior during brainstorming, the participants of the evaluation study deplored missing private space to develop ideas on their own and limited space on the tablet devices in common, since there was only the possibility of global page changing.

NiCE Brainstorming also uses external devices to contribute ideas to a common pool. However, in contrast to the presented system, there is no mirrored canvas and, therefore, no way to interact directly on the interactive whiteboard by using an external device. We believe that this separation helps to increase the number of contributions, because ideas can be developed in private space, aside the public screen, without general attention. Moreover, ideas can be dismissed before publishing, if the creator is not happy with the result. This provides additional safety to try out something new and helps to overcome the fear of evaluation apprehension.

2.5 Presemo Brainstormer

Presemo Brainstorm [20] is a web-based EBS that maximizes the benefit of an electronic brainstorming session by guiding the group through a very tight workflow consisting of different brainstorming phases.

The fact that face-to-face brainstorming is still the dominant type of brainstorming in professional life is a result of the inadequate integration of new electronic brainstorming tools into established practices. Furthermore, studies tend to focus on idea production only, although this part cannot be separated from other phases, such as problem framing or idea selection.

Presemo Brainstorm was designed to guide users through an optimized process of brainstorming. After framing the problem within the group, all participants think about ideas to solve the problem on their own. Then the ideas of the others are unveiled to use associations for further idea generation, followed by a group brainstorming phase. After this, the group selects one idea and develops it further in a group ideation and negotiation phase. Overall the Presemo Brainstorming process starts with the individual ideas and continuously opens up to brainstorming within the whole group. Since the system is web-based, every participant usually uses a laptop or any other browser-equipped device. All ideas are submitted anonymously, following the principles of electronic brainstorming. An evaluation process stated a positive general impression of the system, but participants were unsatisfied with the anonymity within the system, since they tried to resolve identities. Moreover, they claimed insufficient transparency of the whole process and the restriction to textual input only.

The most interesting finding in this paper is the conclusion after analyzing the current dominant position of face-to-face brainstorming. While Hilliges et al.[16] argues that there is a benefit in face-to-face brainstorming that cannot be measured, the developers of Presemo Brainstorm state that the current EBS systems are just not good enough. So, they designed a very tight workflow to provide users with a highly productive tool to generate ideas. However, participants' statements on the subject of anonymity and about the missing transparency of the brainstorming process indicate that there are certain implications of verbal brainstorming people miss during an electronic brainstorming session.

In contrast to the tight scheduled process of Presemo Brainstorm, NiCE Brainstorming tries to provide a system where users can brainstorm as freely as possible. While Presemo Brainstorm is browser-based and does not require any special hardware, NiCE Brainstorming is an approach to provide an interactive environment using novel interface and interaction techniques. So the two concepts are more or less the exact opposite of each other.

Chapter 3

Concepts

3.1 Brainstorming Concepts

3.1.1 History of Brainstorming Techniques

In 1953, Alex F. Osborn published the idea of brainstorming in his book *Applied Imagination* [24]. Osborn proposed four guidelines to frame a group creativity technique:

- focus on quantity,
- withhold criticism,
- welcome unusual ideas, and
- combine and improve ideas.

Since then, the issue has been discussed in controversial ways [35,37]. The main idea is that a group is more than its individuals, because members are able to build on the ideas of others. This assumption is connected very tightly to the first rule *focus on quantity*. It is hard to find good ideas, but if there is a larger pool to choose from, it is more likely that there are some good ideas in between. Moreover, more ideas (even bad ones) give the other group members more material to build on. Since there are always social implications within a group, the rules *withhold criticism* and *welcome unusual ideas* are stated. Ideas are very personal in some way; therefore, people might be afraid to be laughed at, if they come up with a strange thought. Despite these guidelines, many researches criticized the approach, because of this social implication and because of other drawbacks too.

Verbal Brainstorming

Osborn's idea of brainstorming is based on verbal or rather face-to-face interaction within a group. The verbal communication method restricts the group to provide only one idea at a time, while all the other members have to listen. This might be stimulating in terms of creating own ideas based on the ideas of others, but it is also distracting [7, 25]. Beside the so-called *production blocking*, other objections regard *social loafing* and concerns about how others perceive ideas (*evaluation apprehension*). Due to the inadequateness of verbal brainstorming, *nominal brainstorming* [37] was developed to provide an alternative that solves the stated problems.

Nominal Brainstorming

The idea of nominal brainstorming is that people are writing down their ideas without having any exchange about their thoughts. Due to this approach, multiple ideas can be expressed at once. Moreover, it is easy to submit the ideas anonymously. That way, social implications are avoided. Experiments showed that nominal brainstorming generates 65% more ideas, considering the same group size and amount of time, than verbal brainstorming [25]. Nevertheless, verbal brainstorming is still the dominant type of brainstorming, and when people speak of brainstorming, they usually refer to it [1, 7].

Electronic Brainstorming

With the upcoming availability of computers, electronic brainstorming was developed. The goal was to combine the advantages of verbal and nominal brainstorming [7]. Members interact using computers, so they can express their ideas all at once without waiting and gain the high output in terms of productivity that is provided by nominal brainstorming. Since all computers are connected, they can build on the ideas of others. This way, there is also the synergy effect of face-to-face brainstorming. Paulus and Brown described the effect that people tend to write down what comes to their mind at first and later use the ideas of others to get inspired [25]. To avoid social implications as they occur in verbal brainstorming, ideas can be submitted anonymously. Alike nominal brainstorming, electronic brainstorming works in a distributed way and in addition, it also offers all data in digital form without further effort.

The Best Technique—A Matter of Group Size

Electronic Brainstorming seems to be the perfect way of brainstorming (see table 3.1), but, in fact, it is hard to come up with concrete numbers about productivity to compare the techniques, because it strongly depends on the group size [8, 11]. While there is only a small difference about verbal brainstorming being the least productive, the gap between nominal and electronic brainstorming is a lot smaller. Dennis and Valach [8] estimate that nominal brainstorming is the better choice for groups of eight or smaller, since there is no significant advantage but more effort in using electronic brainstorming. For groups larger than eight, electronic brainstorming is the best solution in terms of productivity measurements.

	Verbal	Nominal	Electronic
Synergy	✓		✓
No Production Blocking		✓	✓
No Evaluation Apprehension		✓	✓
No Social Loafing	(✓)		

Table 3.1: Comparison between the different types of brainstorming regarding the occurring group effects.

Performance Estimation

However, despite research results confirm nominal or electronic brainstorming to be the supreme techniques, numerous of companies still use old fashioned verbal brainstorming. This leads to the question if there is anything electronic brainstorming cannot provide, if it is too hard to use, or if verbal brainstorming is just overrated. In fact, there is a strange effect regarding the ability for recognizing performance that confirms this thesis. People tend to overestimate their performance in verbal brainstorming, while they underestimate the efficiency of nominal brainstorming [34]. This raises the question if people are just bad in estimating their performance, or if there are other benefits of verbal brainstorming that cannot be measured as productivity in terms of non-redundant ideas per group. Nijstad et al.[22] ascribes the effect to the perception of continuous activity within the group. People become frustrated when the are working alone and do not come up with as many ideas as they expected, but within a group people either feel that the group is making progress or that it is hard to come up with ideas for others too.

3.1.2 Brainstorming in Practical Use

Most studies on the efficiency of brainstorming took place in laboratory settings, observing participants who did not know each other very well. They often were not diverse in their profession and generated ideas on a subject that would never be realized [25]. Compared with a brainstorming meeting at IDEO, a large product design firm in the US that is famous for creativity and widely known for their brainstorming process, these conditions are poor. There are tremendous differences between a verbal brainstorming under laboratory and production conditions. This are the results of a study performed by Sutton and Hargadon [34] from the Stanford University, who pointed out the major advantages of verbal brainstorming for a firm like IDEO.

Brainstorming at IDEO

Brainstorming at IDEO follows the rules of Osborn fairly closely, but the circumstances differ a lot from those in most experiments. While the rules are the same, people who attend brainstorming sessions at IDEO have past and future task independence, past and future social relationships, a use for their ideas, adequate technical expertise, complementary skills and expertise in brainstorming [34]. This leads to the assumption that brainstorming within a group is more than a simple technique, ready to use, but a skill that has to be trained and used cleverly. The team has to be diverse to get different views on the problem, but the members also need a common knowledge base. Consequently, everybody knows what the other is talking about and even more important people know the others level of knowledge. Therefore, it is clear which issues need further explanation due to continuous social relationship. Moreover, due to the heavy usage, people become experienced in the process of brainstorming. Beside these factors that lead to successful brainstorming the report also points out other motivations for a firm to use verbal brainstorming within a face-to-face group. Especially in complex tasks the entire implications of a design decisions often cannot be considered by a single person. This argument reminds of findings in the area of *group intelligence* [33]. It is necessary to point out that effects of group intelligence require *independent thought* and moreover, *superior judgment*. Since these criteria do not exist here, the effects cannot be lead back to group intelligence.

However, a group of people with complementary background is able to address these problems from different points of view. In addition, for a company as a whole, verbal group brainstorming is also a good way to support the organizational memory of solutions and provide skill variety for employees because of their knowledge exchange within the brainstorming meeting.

Reasons of Using Verbal Brainstorming

There are still many practical arguments for a company to use verbal brainstorming. However, there are also various theoretical investigations that support the assumption that brainstorming is about more than just producing a maximum of ideas in short time. Based on the findings of Sutton and Hargadon [34] at IDEO, Dennis and Reinicke [7] state that verbal brainstorming - despite all its inadequateness - enhances the group *well being* because of its social component. People who attend a verbal brainstorming session refer it with fun and consider it as some sort of vacation within their work day. They change to a different location that provides a relaxing and creative atmosphere, where they socialize and develop relationships while being imaginative and acting in a group. The empirical investigation of Dennis and Reinicke [7] showed that participants perceive no improvements in the task performance by using nominal or electronic brainstorming instead of verbal brainstorming. This indicates that people measure task performance in a different way than just counting the number of generated ideas. The results suggest that verbal brainstorming is the better choice for supporting group well being and providing member support.

3.2 Optimized Brainstorming Process

Analysis of the pros and cons of the presented techniques lead to the key requirements for an improved brainstorming concept. Alongside the benefits on *group well being*, offered by verbal brainstorming, the advantages of electronic brainstorming in terms of *higher performance* and the capability of *distributed* and a*synchronous use* have to be considered.

3.2.1 Overall Concept

These considerations suggest combining the advantages of verbal and electronic brainstorming [16]. Recent findings [7, 34] and the prevalence within the industry indicate that face-to-face brainstorming is the technique supposed to provide the core of such a combination. In order to integrate the advantages of electronic brainstorming, it is helpful to conduct the whole process in digital form.

The Cycle of Brainstorming

Sutton and Hargadon [34] mention another notable detail. In a productive environment, most topics are not resolved after a single session. It often needs multiple iterations to come up with a profound solution to a problem. Although it is not clear if the better way is to start on a blank surface again or if it is the better choice to bring up all the ideas from the previous session, thinking of this scenario a digital solution that includes storing and restoring brainstorming sessions, provides a significant advantage and the freedom to choose. Moreover, the concept of iterations unleashes the full capabilities of electronic brainstorming. If people come up with new ideas between the actual face-to-face sessions, they can contribute their ideas because of the features that enable distributed and asynchronous use. When participants of the brainstorming team have some spare time (e.g during traveling), there is no need to wait. They are able to have a look at the pool of contributed ideas and to add new ones.

Phases in Brainstorming

However, the entire process of idea generation consists of multiple phases, among others including framing the problem, generating, selecting and evaluating ideas. Osborn [24] described three stages of the process:

- fact finding (exploration stage to frame the problem),
- idea finding (actual brainstorming stage to generate a large number of ideas), and
- solution finding (evaluating and selecting stage).

These three phases are part of nearly every brainstorming session within a productive environment. They are also included into Presemo Brainstormer [20], where they are connected with each other, following a tight schedule, to provide an optimal brainstorming process.

Flexible System vs. Strict Sequence

Comparing this idea of a predetermined sequence with the cyclic sessions observed at IDEO and brainstorming performed at other facilities, illustrates the huge gap about what people mean, when they refer to brainstorming. This is what exacerbates the development of an optimal brainstorming process. So the alternative of building a system that restricts to a predefined process is to develop a highly flexible system that allows to adapt the sequence to the needs of the actual session. Of course, it implies high responsibility to interact with a system that does not specify the process of its use. However, the use of traditional mediums does not predetermine the process of verbal brainstorming either, and maybe flexibility is one of the reasons for its success.

In consequence, we believe it is necessary to provide a tool, flexible in use, to supports different varieties of use. Figure 3.1 provides an overview over the different phases within brainstorming and illustrates potential sequences. Each session starts with framing the problem. This is not necessarily a collaborative action, but even if it is not, the other participants have to be informed about the problem or rather the issue in an initial meeting. In many cases, this kickoff meeting is immediately followed by a collaborative, face-to-face brainstorming session, although research indicates that starting with an individual session, indifferent if it is distributed or not, would be the better choice in terms of high productivity. Howsoever, in the proposed process this decision is up to the user. It is feasible to perform multiple collaborative sessions, to enable the idea of cyclic sessions. Between these face-to-face meetings, participants are encouraged to contribute their ideas by using distributed and asynchronous usable mediums and devices. This way, others can build on these ideas, even before the next meeting.

Sooner or later the existing ideas are evaluated, and those to be realized are selected. During the evaluation process or the selection process that often go hand in hand, in a discussion, it is unlikely but possible that people jump back into the phase of idea generation because they are dissatisfied with the existing alternatives or suddenly enlightened with another great idea.

This analysis substantiates the need of a flexible system that combines verbal and electronic brainstorming in a way that enables switching between the methods by providing of a common digital basis. Although electronic brainstorming seems more as a supplement within this scenario, it is an integral component and provides additional value to verbal brainstorming.

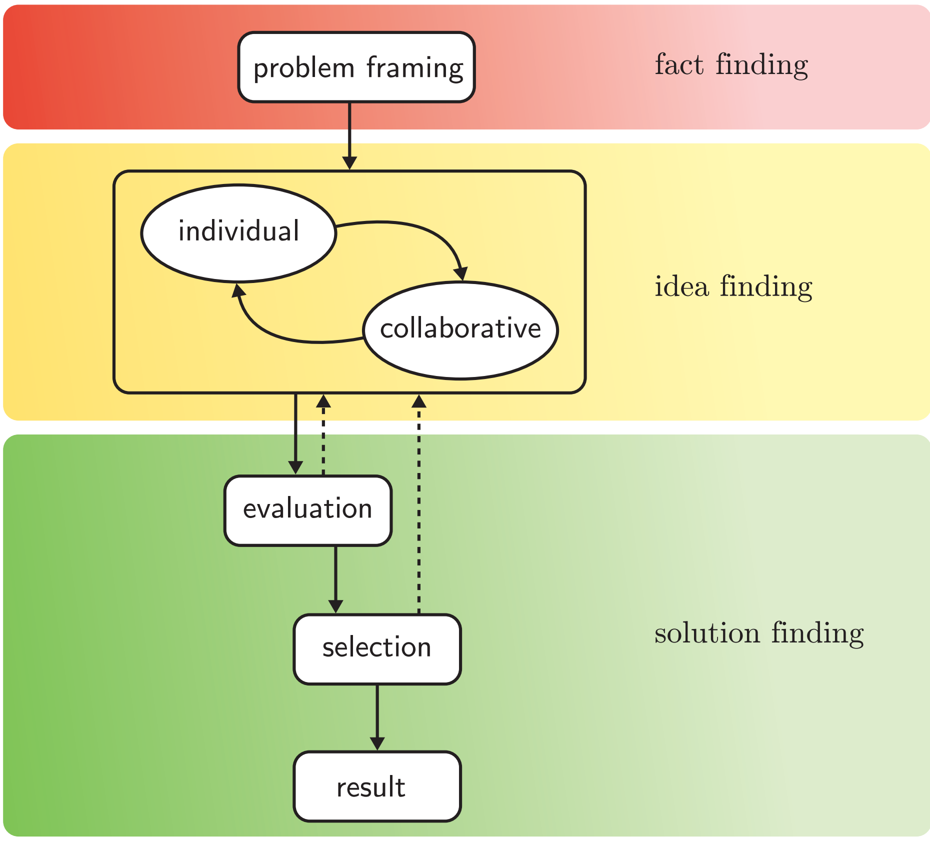


Figure 3.1: Visualization of the different brainstorming phases.

3.2.2 Technical Concepts

Basically a system, capable of handling the specified scenario has to provide two components (see figure 3.2). First of all, an environment has to be created that provides a digital solution for verbal or rather face-to-face brainstorming. An interactive whiteboard with a special software designed for the proposed type of brainstorming seems an adequate device. In a second step, the part that represents electronic brainstorming has to be established. This includes connecting external devices to the system that can be used to submit ideas. Table 3.2 analyses both functionality using the *Groupware Matrix* of Dix [9].

The design criteria for the whiteboard application are predetermined by the traditional verbal brainstorming process, but observing the brainstorming principles used by professionals helps too. The idea is to provide a multi-user whiteboard application that supports fast and fluid interaction in terms of creating and arranging ideas. As the use of physical or screen

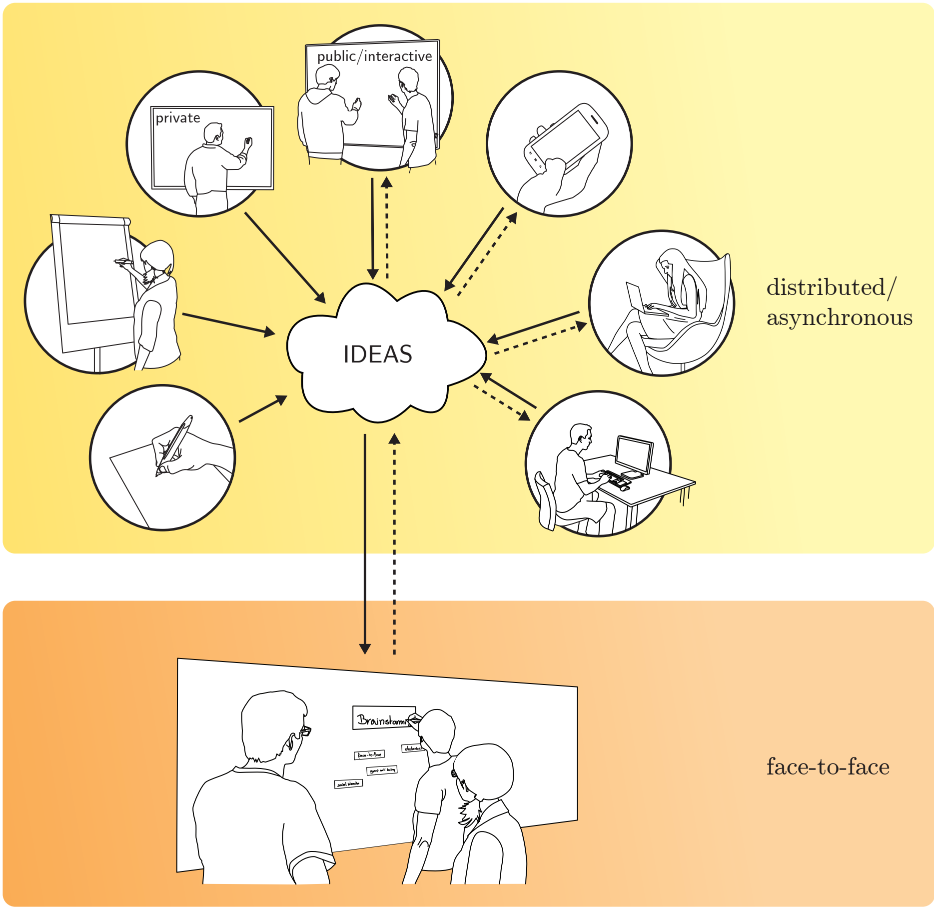


Figure 3.2: Two components of the system: While face-to-face brainstorming sessions are performed on an interactive whiteboard, various devices can be used to contribute ideas in the meaning of electronic brainstorming.

based keyboards is always a dissatisfying solution, handwriting is the way to choose [14, 16].

However, taking a look at the proposed external devices in figure 3.2 displays their difference, also in terms of the content they generate. Writing on a sheet of paper is fairly similar to writing on a whiteboard, regarding the produced type of content. Considering devices, such as computers or Smartphones, it becomes clear that they deliver entirely different types of content. At a desktop computer, it makes no sense to slog away at generating stroke based input using the mouse when a keyboard is available to quickly type the idea. Beside typed words, there are web-sources that might include a novel idea. Thinking of another device, people could use the camera within their Smartphone to capture a photo of something they want to share as a

	Same place	Different place
Same time	Face-to-face brainstorming on the interactive whiteboard	
Different time		Asynchronous and distributed brainstorming using external devices

Table 3.2: The different possibilities to use NiCE Brainstorming, visualized by using the Groupware Matrix of Dix.

cause for thought. In conclusion, the integration of different devices requires to support various types of ideas. Only this way people can fully exploit the capabilities of the different external devices.

3.3 Design Goals

Before starting with a concrete screen design or interaction concept, the main design goals have to be defined. Within this process, three design goals are stated that give a rough outline about the principles behind developing the overall interaction concept:

- multi-user capability,
- fast and fluid interaction, and
- no tool change.

First of all, it is clear to design a multi-user system, to support multiple users in contributing their ideas at once and to enable working in a collaborative way. One of the most essential goals was to create an application that provides *fast and fluid interaction.* Fluid interaction is also one of the main design goals stated by Guimbretière et al.[14]. The importance of this requirement becomes obvious when brainstorming software, designed for desktop systems (e.g. XMind[1], DropMind[2], FreeMind[3], and BubblUs[4]) is observed. Apart from the fact that these programs were designed for desktop systems, they often focus on mind-mapping aspects. This leads to a highly graphical interface where users are provided with tools to make their mind-map look pretty, but without mechanisms to add ideas in a fast and informal way. Especially in a brainstorming scenario it is necessary to enable as easy idea contribution as possible. It is essential to avoid the creation of technical or social barriers [2].

[1] http://www.xmind.net
[2] http://dropmind.com
[3] http://freemind.sourceforge.net
[4] https://bubbl.us

Deduced from this, the requirement *no tool change* is claimed. This includes no menu to change tools but more important, no special state occupied by a single pen. Therefore, the user is not distracted from the brainstorming process by thinking about the pen's state. Furthermore, stateless pens simplify exchanging pens between the users. Stateless means an action always starts when the pen's tip touches the interactive whiteboard and ends when the pen is lifted from the surface. This way the state of the application is altered, but the pen stays stateless. This implies that one and the same item can be used by two users, one after each other or even simultaneously, as long as their actions do not interfere. For instance, a single item can be edited by two users at once, but it cannot be moved by two users at the same time, since this would result in a battle. This example shows that the paradigm does not help to prevent any conflicts that can occur in an ordinary multi-user system. Still two or more users can act simultaneously and so there have to be mechanisms that prevent them from interfering with each other.

3.4 Feature Specifications

Beside these general design goals regarding the overall interaction design, there are multiple features that need to be included into a brainstorming application that is capable of supporting a way of brainstorming like described in section 3.2.1. The predefined features include

- local content creation,
- remote content contribution,
- various types of content,
- content manipulation, and
- structuring.

When performing face-to-face brainstorming, using an interactive whiteboard, it is essential that the software enables *local content creation*. However, a main idea in the presented brainstorming concept is to support a distributed and asynchronous workflow. Deducted from this idea the two feature specifications, *external content contribution* and *various types of content* are stated. To enable distributed and asynchronous content generation, there have to be possibilities to support various types of content. Handwriting is a simple way to contribute ideas through a pen-based interface, like an interactive whiteboard or digital paper, yet other devices require different types of input. Therefore, it is important to have the freedom to choose between different representations, to use the most suitable. There is also no reason to restrict itself to handwritten and plain text, since plenty of other sources like images, links or snippets of websites can be used to represent an idea.

Another fundamental issue is the *manipulation of content*. In order to enable a dynamic flow of ideas, it is not only essential to provide a fast and easy way to generate content but also to manipulate and in particular to remove it. Within a brainstorming scenario, speed is more important than accuracy. Therefore, it is not necessary to include powerful and complicated tools to edit a single idea, but its necessary to include basic features to remove. Finally, *structuring* is a crucial topic. As proposed in section 3.2.1, brainstorming is not only about idea creation but also includes the phase of solution finding. This phase is usually performed in a group, discussing the options while the ideas are still visible for everybody. Mechanisms that support structuring might help to visualize different categories and relations and this way provide additional benefit.

Chapter 4

Application Design

When starting with the development of the overall screen design and interaction concept, the difference between a straightforward sketching application and a brainstorming application was a big issue. A sketching application on an interactive whiteboard is the digital equivalent to a traditional, non-digital whiteboard that is often used for brainstorming. Therefore, there is no doubt a sketching application can be used for digital brainstorming.

The dominant use of traditional mediums, as pen, paper, and whiteboards in brainstorming sessions, is a result from the experience people have in using these mediums but also their flexibility. One of the difficulties about brainstorming is that everybody does it in a different way. Thus, everybody expects something different. There are two opposed strategies to deal with this situation. One possibility is to reduce the features to the lowest common denominator, providing only basic tools. The other extreme is to develop a narrow concept that provides only one way to go and forces the user to use this one method [20]. Of course, there are also different shades in between, but in general the designer has to decide between those two opportunities. Considering the first alternative, maybe a basic sketching application would do the job. Therefore, the question is what features can be added to a sketching application to enhance it for brainstorming. The answer is likely the lowest common denominator of brainstorming.

The following sections comprehend the design decisions that lead to the design of the NiCE Brainstorming whiteboard application. The considerations are based on the feature specifications in section 3.4. In doing so, the content creation process, whether locally or externally, is presented at first, followed by remarks about ways to manipulate the content. Subsequently, various features for structuring are presented, before the overall interface and interaction concept is elaborated. The final section deals with the topic of limited space, its reasons, and how to overcome.

4.1 Content Creation

Within the feature specifications in section 3.4, two methodes of input are listed: *Local content creation* and *Remote content contribution*. The need of a way to create content locally, directly on the allocated interactive whiteboard is obvious, when the goal is to support face-to-face brainstorming. Consistent with the design goals *multi-user capability* and *fast and fluid interaction*, it is necessary that multiple persons are able to contribute content locally at the same time, in an easy and quick way.

However, as described in section 3.2.2, the overall idea of including asynchronous and distributed content contribution, is to support plenty of different devices. These devices differ in the type of content they deliver, but also in terms of their ability to provide feedback. Along several *interactive* devices, as computers, laptops or Smartphones that are able to display dynamic content, there are a couple of *passive* devices as paper, flipcharts or whiteboards. These devices can also provide content digitally, when they are equipped with adequate tracking technology, but they are unable to display any additional material or feedback. The lack of feedback is the biggest issue here, because it is hard to use a device that is not capable of providing any hint, if the intended operations were performed successfully.

Within the design of NiCE Brainstorming, two concepts of creating external content were fleshed out. One is dealing with the use of paper, representing various passive devices like flipcharts or whiteboards. The other concept provides an interface for various interactive devices by using a 3^{rd}-party web-service.

4.1.1 Local Content Creation

At the beginning of this chapter, the question about the difference between a plain sketching application and brainstorming application came up. To justify the use and the development of a brainstorming application that provides a maximum of flexibility within the actual process, it is essential to offer some kind of benefit compared to a sketching application.

Motivation for High-level Interaction

Most brainstorming techniques are characterized by the idea of producing chunks or preferably phrases of information [5, 16]. These phrases usually stand for a single idea or thought. Therefore, they are certainly more high-level than basic strokes. In most cases, they consist of multiple strokes that form words or sketches. While a sketching application usually provides a *Selecting Tool* to select, move or copy a bunch of strokes, it would be a substantial benefit for a brainstorming application to skip this time-consuming process and interact on a higher level.

Problems with Automatic Clustering

While providing a fast and easy way to contribute handwritten content is essential, the comparison with a sketching application points out that additional effort is necessary, to provide high-level interaction. The basic idea is to implement reasonable clustering. Establishing high-level interaction does not provide many benefit for the process of content creation, but it speeds up most subsequent interaction, since there is less need to grapple with low-level activities, as selecting single strokes to move a whole word. In order to provide a fluid experience, the cluster generation has to be performed during content creation. Therefore, the overall goal is to find a mechanism that does not slow down the actual content creation process and produces adequate clusters that are helpful during later interaction, nevertheless. The challenge is to find a suiting algorithm. Data like proximity or creation time can be used, but even then, there is no guarantee for flawless clustering. Most likely, some special cases lead to problems. Working on a semantic level only helps in theory, because it is extremely hard to extract the correct semantic information out of the strokes. Text-recognizing can lead to errors, especially when writing fast and maybe sloppy during brainstorming. Moreover, it cannot help with sketches. However, the biggest problem, when dealing with an automatic stroke clustering algorithm, which is implemented in a sketching canvas, to help the user during a brainstorming session, is that even the best algorithm can fail. In this case, the user's attention is needed to fix the error. This means various additional tools have to be provided to manipulate and to correct the generated clusters, as suggested by Watanabe et al.[38]. Often, this is not a problem, but in the case of brainstorming it distracts the users from their actual activity and reduces the benefits of high-level interaction.

Implicit Clustering

Since none of these considerations fit the needs of a brainstorming application, a different concept has to be developed. When the algorithm cannot provide solid clustering, the task has to be done by the user. However, not to distract the users with an additional task, it is necessary to make the process of clustering as implicit as possible. Whenever users start to write or to sketch, a new visible cluster is created. Now they have the choice to continue writing within the created cluster or to start a new one by writing somewhere else. Since the clusters have to be visible, the metaphor of Post-its is used because they are a recognizable utility in the area of brainstorming. In difference to physical Post-its and to other implementations [16], the clusters used within NiCE Brainstorming are variable in size. The size is dynamically adjusted to the provided content. This decision was also a matter of discussion, because physical Post-its are fixed in size. This limitation might be a benefit, as content has to be short. However, restricting the area to

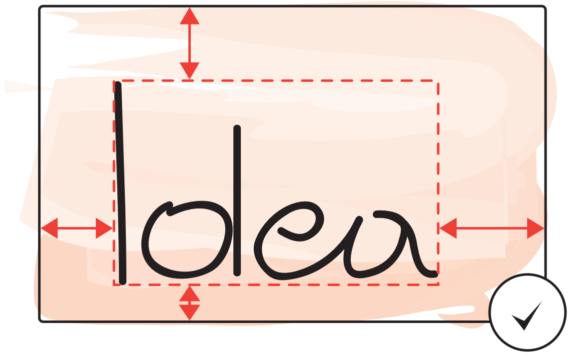


Figure 4.1: The margin at the right is enlarged to disburden fluid writing, while the margin at the bottom is minimized to enable easier line changing.

write to virtual Post-its, fixed in size, leads to frustration if the space is not enough and wastes space if not the whole area is needed. Therefore, the size is adapted in real-time during writing on the Post-it. This enables fluid writing without worrying about any artificial restrictions. The only relevant issue to improve fluid writing is the meaningful definition of the Post-its margin. When testing the concept, it turned out to be helpful to widen the margin at the right to disburden fluid writing and also to shorten it at the bottom to enable quick line changing because a line change usually indicates the begin of a different cluster (see figure 4.1).

Shrinking Handwritten Clusters

After its creation, the cluster stays editable until a new cluster is started or the end of editing is confirmed manually by pressing a button at the lower right of the Post-it. In either way, the oversized margins are removed, and the Post-it is also shrunken down, cf. Hilliges et al.[16]. Shrinking reduces the space ideas require on the whiteboard and helps to address the problem of limited space (see section 4.5). It turned out that shrinking handwritten text to 50% of its original size is usually not a problem in terms of readability, since verbal brainstorming is usually performed within small groups and the participants are fairly close. In order to change back to edit mode, the user has to perform a single tap on the Post-it. It is instantly enlarged to its original size and ready for further editing. Also, the oversized margins are restored.

4.1.2 Use of Paper

Despite countless digital devices that accompany us in every circumstance, many people still use paper as their primary tool to catch ideas or to take quick notes. Therefore, it seems a logical step to provide a way to digitize these notes, to integrate them into the system.

Countless tries were performed to provide a quick way to digitize notes made on paper [4, 19, 41]. Many prototypes and products were designed to connect the whole workflow that is performed on paper to corresponding digital tools. This includes notes from meetings, addresses, telephone numbers and other categories [4]. Compared to these complex tasks, the problem here is simple. The basic goal is to digitize a written idea and to contribute it to a common pool of ideas. Due to flexibility reasons, using a ceiling mounted camera, as Guimbretière et al.[14] or a scanner was not an option. Digital pens seem an adequate device for the job. There are plenty of different types on the market that use different tracking technologies. Dependent on the task, different pens are imaginable. The presented concept regarding the use of paper is not specified to a special kind of pen, since no buttons on the pen or other specific functions are used. The recognition of different pages is the only condition.

Due to the lack of a dynamic interface, many paper-based systems require complex gestures to trigger commands, others use more implicit commands [31]. A questionnaire, which was performed on the University of Economics Vienna among economic students who worked with digital pens on paper during their lessons, pointed out that they felt if they had no or insufficient control over their work. Moreover, within their replies they stated that the use of digital paper was convenient because they could write as they were used to on real paper but that they missed possibilities of error correction.

Digital Paper—Private Notes Go Public

In point of fact, traditional paper can be a very private medium. By digitizing, it gains capability to publish content. This way it becomes public. Many users fear this intrusion into their private space. In addition, they often cannot fully understand what is going on when they use digital paper. In many cases, specific gestures have to be performed to trigger a function [19] or the systems use implicit commands [31]. Even if implicit commands provide a natural feeling and fluid interaction, users have to be aware that they are interacting with a system. Therefore, they have to know about the commands they can use. Other scenarios provide buttons on the paper that can be pressed, by using the pen to trigger different states, for instance toggling visibility on a public screen. However, it is also possible to execute this commands using a separate digital device. Together with missing distinct visual feedback, it is hard to gain any certainty about what is going on. The lack of a channel that gives instant feedback on what is happening during the use of digital paper is the missing feature here. This might not be a problem for experienced users, who are familiar with the system they use, but people who have no experience with the use of paper-based interfaces might act cautious because they fear unintended results. So it is necessary to make all the actions that can be performed with digital paper as obvious to the users as possible, since the only way of losing fear is to regain control over the used tool. This way, users are encouraged to use digital paper and include it into their everyday work.

Requirements for Concept Design

Once taking a look at informal ideas scribbled on paper, it becomes obvious that there are significant differences in the way various people organize their notes. Some use a whole page to sketch a single idea, while others tend to write down short and independent notes, distributed over the whole sheet of paper. In order to overcome these differences, a flexible system is necessary that enables to crop and send content.

Apparently there is a gap between the claim for flexibility and the previous observation that the sequence of operation has to be as simple and obvious as possible to encourage people to use digital paper. In order to fulfill these requirements, two concepts where developed to enable paper as asynchronous and distributed device for brainstorming. Users should be able to write or to sketch their idea and then choose whether to send or not to send it to a specified idea-pool, since there can be multiple projects at once that use the distributed brainstorming concept.

Concepts

The first concept is based on the tagging-metaphor used in NiCE Book [4]. For each project that uses NiCE Brainstorming, a unique sticker is printed and distributed among the participants. There is a blank page that can be used in any way to record ideas (see figure 4.2 (a)). If users want to add their ideas to the public pool, they stick a sticker on the sheet of paper and

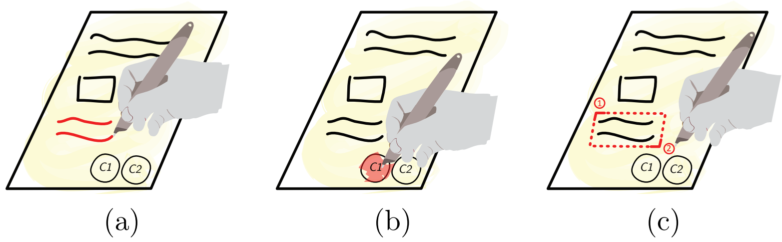
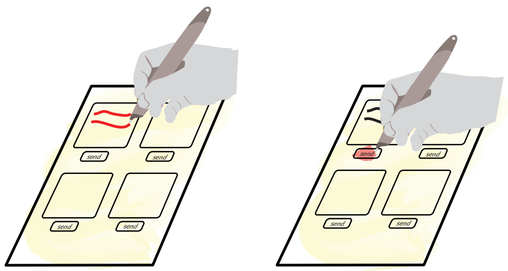


Figure 4.2: (a) The whole sheet of paper can be used for notes and scribbles. The content can be arranged in own terms. (b) To link the content with a brainstorming session and start the contribution process, the according sticker at the bottom has to be pressed. (c) By marking two corners, a bounding rectangle is created and the covered content is sent.

tip on it with the digital pen (see figure 4.2 (b)). Afterwards, the selection is done by marking two corners of the bounding rectangle with the tip of the pen. The submission is confirmed with the second point that defines the bounding box as seen in figure 4.2 (c). This concept is as simple as possible, while providing the necessary flexibility that allows users to use the whole sheet of paper in own terms.

The only way to simplify the process even further is to forgo flexibility. Therefore, the second concept uses different pages for each category. There are predefined areas on each page, with a corresponding button, next to each area. Users can write or sketch their idea into the area and submit it by tipping on the button afterwards (see figure 4.3). The idea is remarkably similar to the use of Post-its, apart from the fact that a couple of Post-its are printed on a sheet of paper with a corresponding button. Anyway, it would be easy to use actual Post-its for this purpose.



(a) (b)

Figure 4.3: (a) Only the dedicated areas can be used for notes and scribbles. (b) The button underneath each area is used to send the content of the corresponding area. There is no dedicated selection of the brainstorming session as there are different sheets of paper for each session.

It is pointless to mention that the first concept is far more flexible. For example, it would be possible to use the similar process to send data from a traditional whiteboard or flip chart if there is a way to track a digital pen on it. However, it may be an illusion that people constantly use digital paper; therefore, the second concept was implemented to provide a possibility to submit ideas from paper. Maybe it is unnatural to write ideas into a dedicated area on a sheet of paper, but it allows straightforward submission without gestures or other implicit knowledge.

4.1.3 Sharing Notes

There are plenty of scenarios where one might come up with a new idea and wants to share it. According to the situation, different devices, ranging from desktop computers to Smartphones and tablets, can be used. Supporting all those different types is not easy.

Various Types of Content

Especially since the integration of various devices does also include the support of *various types of content*, as stated within the feature specifications in section 3.4. Within a face-to-face brainstorming session, handwritten notes and sketches are the commonly used type of content. However, the concept of distributed and asynchronous brainstorming is dependent on external devices that are capable of providing ideas. Types of this content may differ from the handwriting directly written on the whiteboard, due to different input capabilities of the devices. There is no need to "paint" notes by using the mouse, when sitting in front of a computer that includes a keyboard. Moreover, there is no reason why the camera of a Smartphone cannot be used to take a snapshot that visualizes an idea. Therefore, the whiteboard application is supposed to support various types of content, in order to enable the unique features of external devices. Alongside handwritten content, these types may include images, typed text, or web-snippets.

Usually supporting many different devices requires immense effort. Therefore, the use of a common web-based standard that works on all browser-equipped devices, is a good choice. In doing so, the integration of a 3rd-party web-service is reasonable, to minimize the effort even further and gain access to an existing infrastructure.

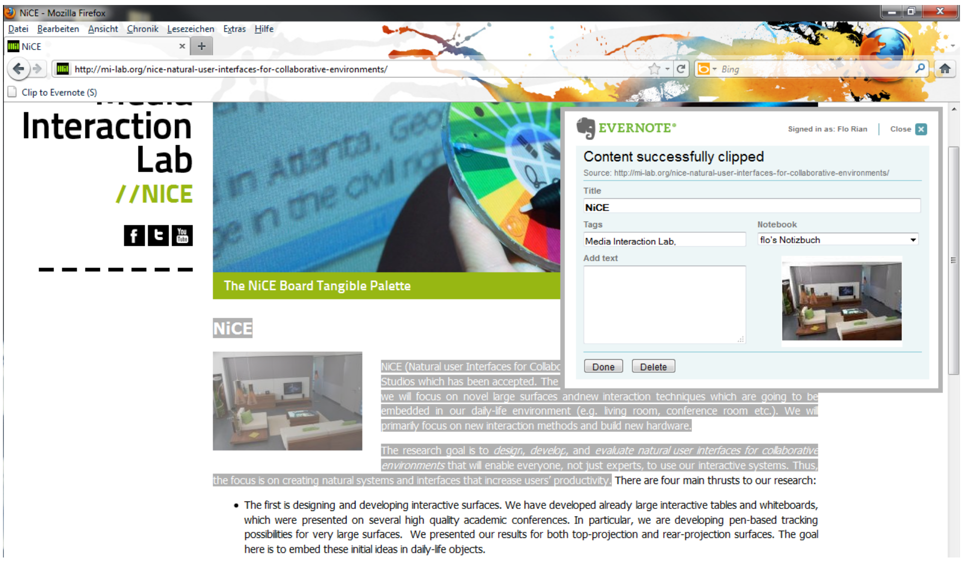


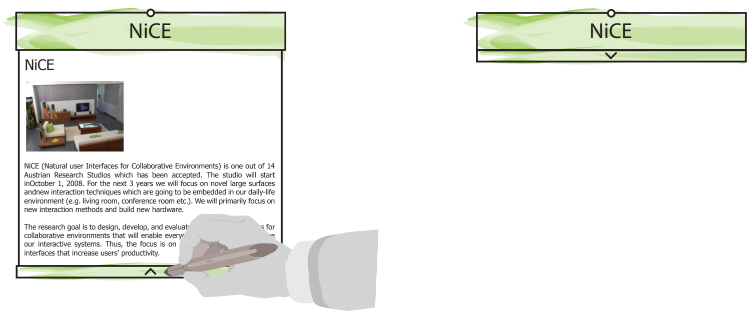
Figure 4.4: A new note is generated from the marked text within the browser using the Bookmarklet.

Evernote

Evernote[1] is a note taking tool that was designed to store ideas from different sources. It was designed to work with any modern computer or mobile device, and it can also be integrated into various browsers as plug-in or Bookmarklet[2] (see figure 4.4). Supporting Evernote covers many devices and enables idea contribution on different levels. This means ordinary text messages or images can be submitted as well as web-snippets that can be generated easily within the browser. The notes are stored in a general (HTML-based) markup-language and can be accessed over the Evernote API.

Displaying Web-snippets

While providing content by using Evernote is quite simple, displaying all the different types of content on the whiteboard is a challenge. Especially web-snippets are difficult to handle because of their size and their diverse structure. Keeping the statements about resolution on the whiteboard in mind, confirms the need to reduce the size of the snippets. However, this interferes with the requirement that the text on these snippets has to stay readably. In order to enable small size and readability of the text, some kind of zooming mechanism is indispensable.



(a) (b)

Figure 4.5: (a) An Evernote web-snipped in its standard visualization. (b) By tapping on the bottom of the viewer it can be collapsed even further to save space.

Therefore, a standard size of these web-snippet viewers was defined. At most they are 200 pixels in width or 200 pixels in height, dependent on the ratio of the snippet. The viewer shows the title of the note and a thumbnail of the included web-snippet. In order to shrink it even further, it can be collapsed by tapping on the bottom of the viewer (see figure 4.5). A tap on

[1] http://www.evernote.com/

[2] A Bookmarklet is a bookmark within a browser that contains no actual link, but JavaScript code that calls a certain service.

the thumbnail enlarges the content to its original size. At the same time, the viewer becomes editable. Handwritten notes can be added by painting them over the web-based content to mark certain areas or to emphasize a passages (see figure 4.6). Moreover, keywords and a button that calls the original web source via the browser are added within the title bar. The viewer automatically changes back to its standard size when another item is touched or when the button at the lower-right of the item abandons edit mode.

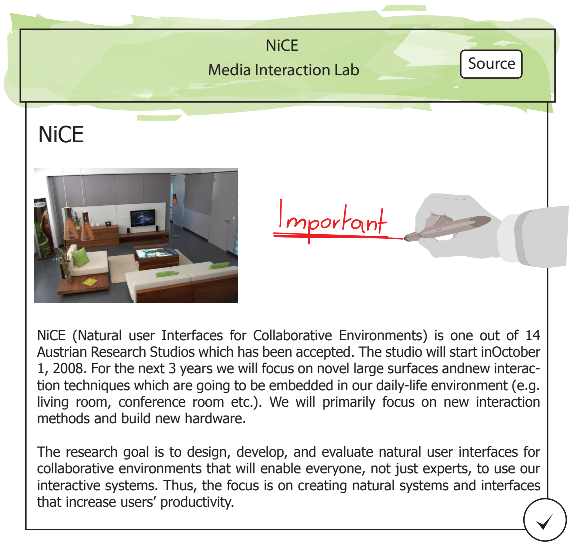


Figure 4.6: In edit-mode it is possible to mark important areas on top of the web-snippet or to add some additional notes. The button in the upper right opens the original web-source within an external browser.

4.2 Content Manipulation

Fast interaction is a key point in brainstorming. There is no need for complex and time-consuming idea editing. In many cases, it is easier to throw away a piece of content and create it all over again. Of course, there are also situations where the possibility to perform simple changes is helpful, for example, thinking of a misspelled word. Therefore, basic *content manipulation* capabilities are essential, even within a brainstorming application.

Content Editing

In a standard sketching application, editing is difficult. The tool has to be changed from pen to eraser; the wrong stroke has to be deleted; the tool has to be changed back again, and not until then the mistake can be corrected.

Since the design goals state that any kind of tool change has to be avoided, but some kind of editing is badly needed, a different solution is chosen. As seen in figure 4.7, a scratch gesture is used to remove stokes. The concept is similar to the NiCE Formula Editor [18], where it performed exceptionally well. The gesture can be used to perform basic edits without changing the tool. Moreover, this gesture can be used on different levels. Besides removing strokes when editing the content of a Post-it, it can also be used to remove the whole Post-it or other objects such as connections.

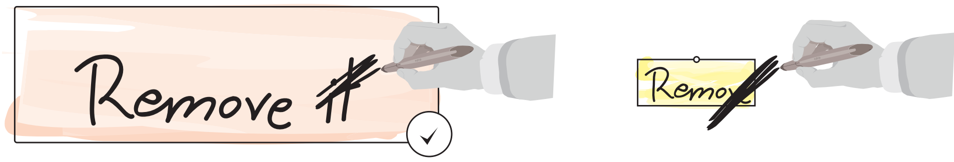


Figure 4.7: The simple scribbling gesture is used to delete content on multiple levels. While on the left some strokes are removed within editing mode, on the right the whole cluster is deleted.

Concept of Metastrokes

The use of strokes for editing interferes with the creation of new handwritten clusters. Hence, whenever the first stroke outside an existing cluster is performed, it is checked to verify if the stroke has a special meaning. If the detection is positive, the stroke is treated as Metastroke and triggers a command. If the detection is negative, a new Post-it is created. Beside the mentioned scratch-gesture to remove strokes, there is only one other command that can be called by using Metastrokes. If one or more items are encircled by a stroke, just alike using a lasso tool, these items are selected. The selection is helpful to move multiple items at once and can also be used to create permanent groups.

The Wastepipe

A method that is in a way related closely to deleting is restoring. Most applications support some kind of Undo/Redo mechanism to restore unwanted actions. However, it is tricky to implement Undo/Redo within a multi-user environment. Compared to the actual needs within the application, it is a concept far more powerful than necessary. The basic need is to restore objects that are deleted by accident. Of course, there is also the possibility of including an additional query to verify the intent of executing the delete command. However, this would result in additional effort for the users, every time they remove some content.

The concept developed to accomplish this missing possibility is called

Wastepipe. Basically it is the idea of a Recycle Bin that collects all deleted items. These items are minimized and displayed in a bar at the bottom of the workspace, since this is usually the place that gathers least attraction from the participants of the brainstorming session. Whenever users want to restore one of the items within the Wastepipe they can send it back to its original position by a single tap or rip it out of the pipe by using an ordinary drag & drop gesture. A special algorithm was designed to prohibit unwanted dragging. It simulates resistance that the user has to overcome to rip the item out of its position within the Wastepipe. Details about the implementation can be found in section 5.4.

4.3 Content Structuring

Though the ability to structure is not essential for the phase of idea generation, it is helpful in the phase of evaluation and selection. Mechanisms that support fast structuring are helpful to arrange content and offer a way to visualize dependencies. As well structured and arranged content is easier to gather and understand, this offers significant benefit, especially in the phase of solution finding. Within this section, four mechanisms that support structuring are presented.

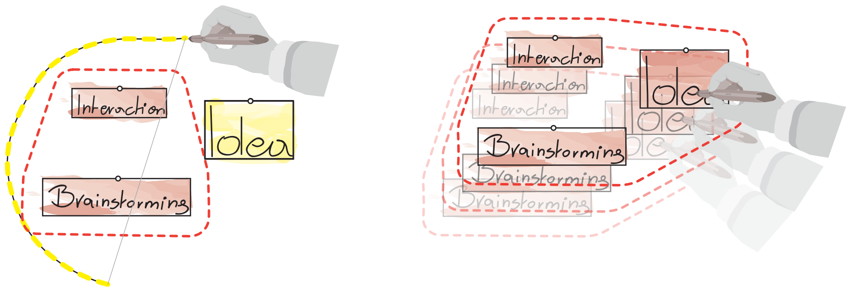
4.3.1 Moving Items

Although it seems to be an extremely simple process, taken for granted on an interactive whiteboard, it is worth to be mentioned. Since the overall interaction concept is based on items that provide chunks of information, it is easy to move them around. However, this process would require much more effort, if there were no clustering that takes the interaction to a higher level. Due to the method of implicit clustering, as described in section 4.1.1, it is possible to implement the issue of moving on a very high level. Tipping down on a specified item, performing a drag gesture and lifting up the pen, is enough to move an item. Of course, moving works in the same way for all different types of items, but only if the item to be moved is not in edit mode.

There is also a convenient way of moving multiple items at once. When encircling several items, a selection is established. This action is similar to the familiar lasso tool. The stroke is interpreted as Metastroke (see section 4.2) when the predicted closed lasso covers at least 50% of one item (see figure 4.8 (a)). When moving one item of the selection, all the other items within the selection are moved in the same way, as seen in figure 4.8 (b).

4.3.2 Connecting Items

Many cases that exceed the traditional process of brainstorming include arranging ideas to visualize their relation [17]. Often, proximity is used to



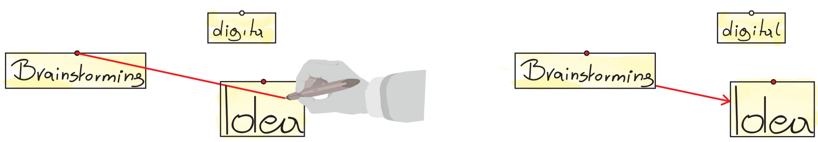
(a) (b)

Figure 4.8: (a) By encircling items with a basic stroke, a lasso selection is performed and the items that are covered by the lasso are added to the selection. (b) After they have been selected, multiple items can be moved at once.

indicate relations. Special features like connecting or grouping are supposed to support this process.

Connections help to visualize relations between different items. Since the *no tool change* paradigm (see section 3.3) prohibits the use of a specific *Connecting Tool*, another way was established to connect items. Each item that is capable of being used within a connection has a small pin at the top. As seen in figure 4.9 (a), this pin provides a starting point to connect the item with another one. Thinking of a physical metaphor it can be compared with fixing a piece of twine between two pins. Although the metaphor is not entirely visually correct, since the starting point of a finished connection is not the pin but the center of the item (see figure 4.9 (b)), this algorithm was chosen due to its clear and clean appearance. Every connection has a dedicated origin and a target. The orientation is also visualized by an arrow. A possibility to establish a connection between dedicated equal items is to create a second connection in the opposite direction.

A big advantage of this type of connection is that the system is aware of the items that are connected. This information can be used to update the



(a) (b)

Figure 4.9: (a) A connection between two items is established by connecting the pin of one item with the other item. (b) An established connection is visualized by an arrow.

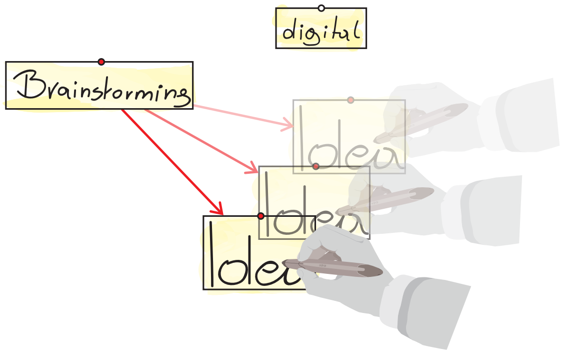


Figure 4.10: When the items are moved the connection is adapted.

connections when items are moved (see figure 4.10). If visualizing connections were only possible by drawing on some kind of meta-canvas, this would not be possible. Moreover, it would interfere with the standard Post-it generation. Of course, it has to be possible to remove connections. For this purpose - once again - the scratch gesture is used.

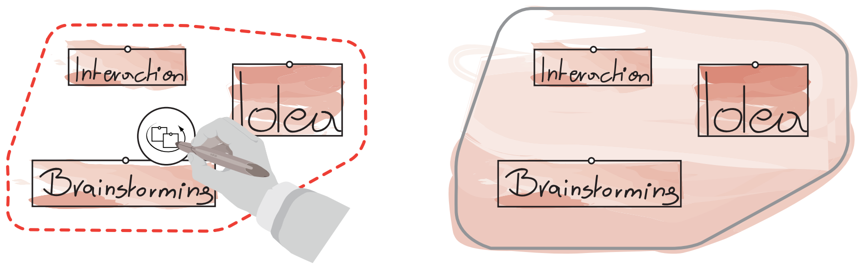
4.3.3 Grouping Items

Arranging items next to each other to indicate relationship is a common approach. Groups support this concept and provide additional visual and operational opportunities. To establish a group, a selection of items has to be created. As described in section 4.3.1, this is done by drawing a lasso around the items that are supposed to be selected (see figure 4.8 (a)). Any set of items can be selected, but only if the selection meets two criteria, a group can be established:

- More than one item has to be selected, and
- no item is allowed that is already part of a group.

If these criteria are met, a button appears in the middle of the selection that enables turning the selection into a group (see figure 4.11 (a)).

Originally a second way to generate groups was implemented. The idea was to drop an item onto another one to form a group. In order to provide feedback about this process, a preview group was generated when one item was moved above another one. However, testing this approach resulted in the comprehension that constant generation of preview groups is too much distraction, especially during moving items all across the whiteboard and since the creation of groups is an infrequent task, the higher cost is adequate. Therefore, the second possibility is now disabled. More or less, this approach creates groups on the proximity of items. Problems with this concept were also observed by other researchers [14, 16, 17].



(a) (b)

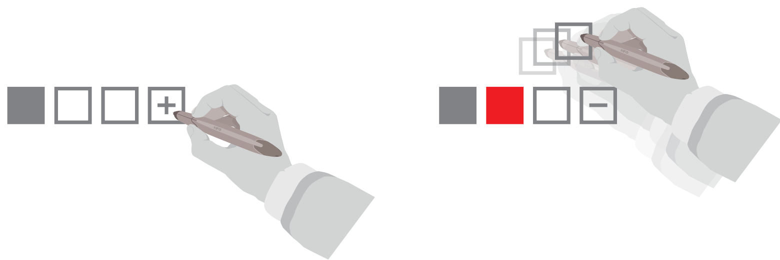
Figure 4.11: (a) If the selection fits all necessary criteria for the creation of a group, the grouping button appears. (b) When the group is confirmed, the border gets solid and the relation is indicated by the color.

Within a group, the color of the items is changed to indicate their relation. Moreover, the whole group is covered with a convex hull of the same color (see figure 4.11 (b)). The hull also provides a menu to change the color to provide visual differences between various groups. The whole group can be moved at once by touching the group's background between the items. This provides rearranging the items within the group by moving a single item as it would be done outside the group. Adding an item to the group is easy. As soon as it is moved toward the group's border, it is instantly covered and becomes part of the group. Removing an item from the group is done in the opposite way. The item has to be moved outside the hull that covers the group. In order to allow fluid rearrangement within the group, the hull is constantly adapted to cover all items of the group. To enable the possibility to remove items from the group, this adaption is delayed. If the speed is high enough, it is possible to rip an item out of the group; otherwise, the hull is adapted and the item is only rearranged within the group. Details about the implementation can be found in section 5.3. When all items are removed from the group, the group is deleted.

To save space, or to focus on a limited set of items, it is possible to collapse a group. When performing a single tap on the group's background, all items within the group are collapsed to a stack. This stack can be moved around like any other item. Another single tap on the stack opens the group again and restores the previous arrangement of the grouped items.

4.3.4 Instancing the View

Instances serve as virtual whiteboard extension. When creating a new Instance, a copy of the whole current screen, including all visible items is



(a) (b)

Figure 4.12: (a) To create a new Instance the button labeled with plus has to be pressed. (b) To remove an Instance, it has to be ripped out of the panel and placed on the button that is now labeled with a minus. The red square is a placeholder to visualize where the Instance was. If the pen was lifted now, it would jump back to this position.

constructed. This mechanism is helpful in various situations. First of all, the copy can be used as a snapshot, to store a particular state within the brainstorming session. The mechanism can also be used to create new "pages" that provide additional space. This includes removing all copied items. However, the most common usage is to generate a copy of all the available items to arrange them in a different way. The idea is very similar to the concept of branches as it is used in version management systems. Instances provide a possibility for the participants of the session to become aware of different relations between their ideas without loosing the current structure. The purpose is to offers new perspectives.

As seen in figure 4.12, the control panel of the Instances is fairly basic. Each Instance is represented by a small square in the lower right of the whiteboard surface. While the current Instance is visualized by a filled square, all the other inactive Instances are visualized by outlined squares. By tipping on the symbol that represents an inactive Instance, it becomes active. New Instances can be created by using a button labeled with a '+'-symbol (see figure 4.12 (a)). To remove an Instance it has to be ripped out of its actual position and dragged back on the creation button, whose label now turns to a '−'-symbol (see figure 4.12 (b)).

Although the mapping between Instance and its representation is only supported by the position of the square, this visualization is adequate in practical use. An earlier implementation used thumbnails with a minimized view to represent the whole screen and give a hint about the actual content of the Instances. This method offered better results in terms of recognizing a particular Instance but also required far more space, and since, in most cases, there are only few Instances that are easy to remember by their position, a slim visualization was the better solution.

4.4 Interface and Interaction Design

The overall interface and interaction design is a result of the concepts described in the previous sections. The designs were developed on the basis of the feature specifications from section 3.4 and the design goals from section 3.3. Especially the design goals had a great impact on the interaction design and thereby also shaped the appearance of the interface.

4.4.1 Object-Oriented User Interface

The idea of *Object-Oriented User Interfaces* (OOUI) was created in the early days of graphical operating systems [36]. In contrast to console applications, where an abstract command is used to execute an action, it became possible to display a virtual representation of the desktop on the screen and link applications with familiar icons. These icons are considered as objects that represent a certain functionality of the computer.

Within his book *Designing Object-Oriented User Interfaces*, Collins proposes a set of characteristics for OOUI [6]. His ideas are very much based on the findings of Tesler [36] but describe the idea of OOUI in a more general way. The set includes:

- users see objects and choices displayed graphically,
- the syntax of commands is "object-action",
- users get immediate feedback from actions,
- the interface is modeless,
- the interface displays objects in WYSIWYG[3] form, and
- objects and actions are consistent.

Comparing the design goals from section 3.3 to the characteristics of an OOUI, illustrates that the use of this concept to realize the application is a perfect match. First and foremost, an application designed in accordance with the principles of OOUI can always be capable of providing *multi-user capability*. Since all the commands are "object-actions", there is no interference when multiple users are interacting at the same time, if they are interacting on different objects. Interaction on the same object is also imaginable but requires explicit mechanisms for conflict avoidance. Due to the strong object-orientation of the interface in the NiCE Brainstorming whiteboard application, this is of minor importance although some items support multi-user interaction. For instance, multiple users are able to write on a single Post-it all at once, just as in reality.

Another design goal that is implicitly solved by the use of an OOUI is *no tool change*. As all commands are applied directly on objects, it is easy to avoid the integration of tools. That is why there is also no need for a

[3] What You See Is What You Get

menu that offers the possibility to change tools. This is especially helpful in a multi-user environment that would usually require multiple menus to make them easily accessible for all users. These remarks do not cover the use of any "global" menu to offer opportunities to save or to load sessions.

The following paragraphs illustrate the realization of OOUI characteristics within the NiCE Brainstorming whiteboard application.

Users see Objects and Choices Displayed Graphically

Taking a close look at the interface and interaction concept and applying the characteristics of the OOUI concept, unveils the object-oriented manner of the whiteboard application presented within NiCE Brainstorming. Visualizing items, using the familiar shape of Post-its provides an impression of the opportunities in terms of interaction. For instance, every item can be moved, just like any Post-it in reality. Another visualization of options is the pin at the top of each item that serves as a handle to connect items. However, there are also more subtle ways to interact with items, thinking of using a tap on the item to open edit mode or the use of gestures.

The Syntax of Commands is "Object-Action"

The overall interface and interaction design is unquestionably object-oriented. Every interaction affects at least one item and thereby provides a feeling of direct manipulation. Even when users start to write on empty space, a new item or rather object is instantly and implicitly created. Also, the scratch gesture to remove and encircling to select can only be performed in combination with existing items. As a consistent result, performing these gestures without any items involved, results in the creation of a new item with a stroke that represents the gesture on it.

Users Get Immediate Feedback from Actions

Each command and effect is immediately confirmed by a visual conversion of the item. Especially when using input devices such as pens that provide no haptic feedback, it is crucial to confirm each action visually and provide instant feedback on what is happening. Within the application, there are various forms of visual feedback. When leaving edit mode, the color of the item is changed, and it is shrunken down. Also, when starting an action to connect two items, a red line is displayed from the pin to the tip of the pen, until the pen is lifted. Moreover, the pins of the items to be connected turn to red. Another example is the display of an overlay, when encircling items to select them. As soon as a valid selection is established, a preview is visualized, and the user has the choice to lift the pen to confirm or to invalidate the gesture.

The Interface is Modeless

The current version of the application is modeless. Neither the application nor a pen is capable of containing a state. While it was a design goal to avoid different modes for pens because of the reasons described in section 3.3, it is not entirely excluded that the application and this way the interface, stays modeless. Since brainstorming consists of multiple phases as described in section 3.2.1, it might be helpful to implement different modes to enable better support for each phase (see section 6.1.4). However, this would imply a break in terms of providing an OOUI and, moreover, a challenge regarding the visualization of different modes and changes in behavior.

The Interface Displays Objects in WYSIWYG

This paradigm is mainly relevant when the application is used to produce content for printing. As the whiteboard application is considered as part of a digitized working-environment, printing is not a big topic, although further processing of the created content and export to other formats is certainly an issue to be considered in the future. A more general interpretation of WYSIWYG states that changes on the object have to be instantly visible, what is clearly the case in the presented application, although edit mode of items may be a matter of dispute here.

Objects and Actions are Consistent

By providing the same interaction techniques for all the items on the screen, this requirement is also fulfilled. Of course, there are differences about what can be done with different items, but there is never an action that produces a different result on two different icons. The worst thing to happen is that an action produces no result because the item is not capable of the behavior.

Overall speaking, the application provides an interface and interaction design that inherits many ideas of an OOUI. The result justifies the analogy and offers a meaningful validation of the interface and interaction concept.

4.4.2 Global Interface Design

Thought the interface mainly consists of loose components and items, there are some elements with a fixed position. Wastepipe, Mailpipe, and Instances are these globally fixed elements. Figure 4.13 provides an overview over the interface, the global elements, and some of the items. To gain more space it is possible to shift the Mailpipe upwards by pressing the button in the middle. The Wastepipe is not annoying, since the bottom usually gains little attention. Due to the small size of the visualization, the Instances, displayed at the lower right, are generally not a problem. The effectiveness of global

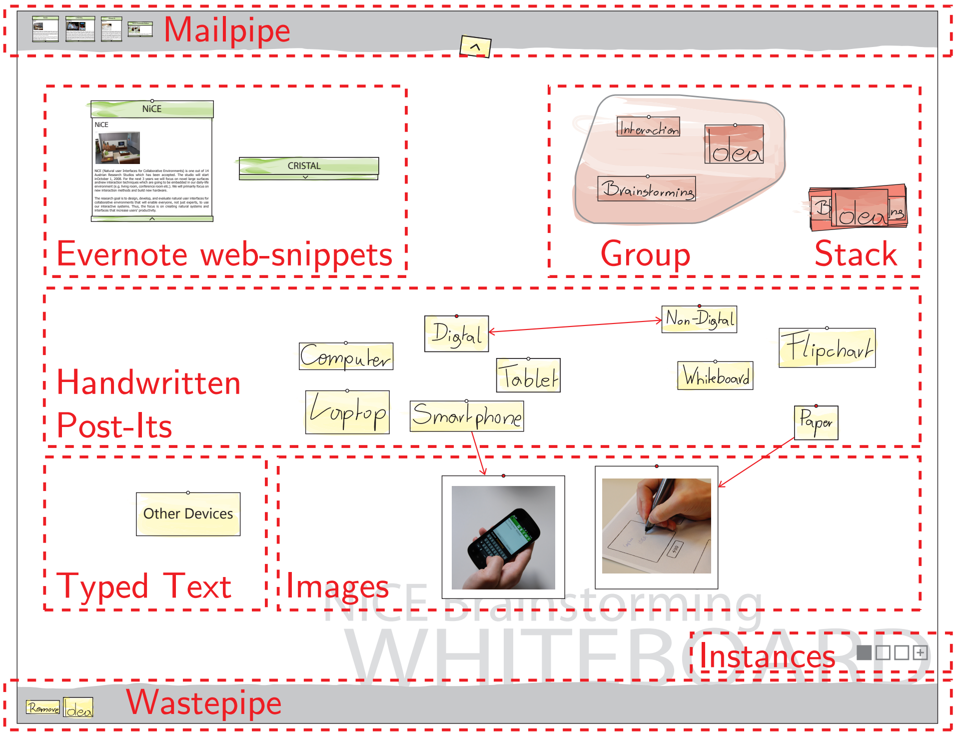


Figure 4.13: The interface of the NiCE Brainstorming whiteboard application: The items, including Evernote web-snippets, handwritten Post-its, typed text and also images can be placed and moved at will. By using the pins, connections can be established and items can be merged in groups that again can be collapsed to a stack. Wastepipe, Mailpipe and Instances are fixed in their position.

elements is connected exceptionally tight to the overall size of the screen. On a single screen setup, the use of global elements is no problem. Moreover, they disburden the interaction in many ways. However, the larger the screen, the more difficult it is to use them in a reasonable way.

Beside these fixed elements, figure 4.13 shows a number of movable items and some of the techniques presented in section 4.3 too. Four main types of items are illustrated, including Evernote web-snippets, handwritten text, typed text and images. While the web-snippets that are visible within the working area, are downloaded from the Evernote server and appear in the Mailpipe, typed text and images are dragged into the NiCE Brainstorming application from 3rd-party applications, using Drag & Drop. This mechanism is reasonable on large whiteboards, where multiple applications are running side by side. However, there are also other opportunities to contribute these types of content, for instance, using the *NiCE Brainstorming Satellite Client* (see section 5.1).

4.5 Handling Limited Space

Every whiteboard, whether it is interactive or not, only offers limited space. While, on a whiteboard, the only possibility to generate new space is to remove content; space on interactive whiteboards can be extended virtually. However, on an interactive whiteboard the resolution is a critical factor that sets a limit to the content that can be displayed at once. The more resolution is restricted; the more important is an efficient space-management. The NiCE Brainstorming whiteboard application uses no infinite canvas that provides infinite virtual space; nevertheless it offers a couple of mechanisms to overcome the problem of limited space, on the one hand, by reducing the size of the content and, on the other hand, by providing mechanisms to hide content or extend the available visual space.

4.5.1 Shrinking Handwritten Ideas

People tend to write much larger on whiteboards than it is actually necessary. This might have to do with the frequent use of writable vertical surfaces, such as whiteboards or blackboards, in front of a larger audience, where it is essential that even people in the back row can read the content. Another possible explanation is the challenge in performing extended precise motions on vertical surfaces, where it is more difficult to get the hand fixed than on a horizontal surface. In his dissertation, Guimbretière [13] points out that people write about 4 times larger on an interactive whiteboard than on a table. Due to the use of a high-resolution whiteboard, as the Stanford Mural [40], it is possible to downsize handwritten content to 25% of its original size, while the content still remains readable. Within PostBrainstorm [14] a dedicated area, called ZoomScape is provided, where discarded content is downsized to 25%.

Due to limits regarding the resolution, we found out that approximately

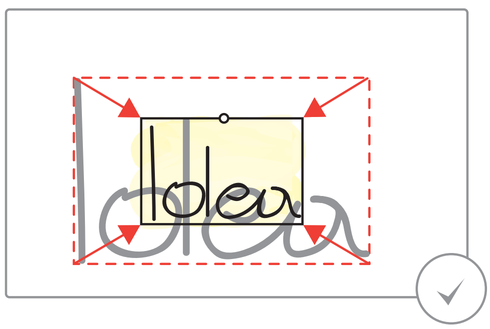


Figure 4.14: When editing mode is left, the oversized margins are removed, and the Post-it is shrunken down.

a reduction of 50% is reasonable (see figure 4.14) when using a low-resolution setup. However, unlike the use of a dedicated area for reduction, handwriting is only showed in its original size during editing, while otherwise it is shrunken down [16, 21]. This way a difference between size to edit and size for permanent display is provided. Moreover, the oversized margins that support more fluid writing are removed. Usually it is no problem to read the notes from a couple of meters away, what is suitable for most scenarios. Observations showed that users felt a little bit awkward at first, when their words were shrunken down automatically, but soon they enjoyed the freedom of writing in a comfortable size and having plenty of space nevertheless.

4.5.2 External Content

Including external content suffers from two different problems: *limited space* or rather *limited resolution* and the *problem of placement*. The first issue is illustrated in figure 4.15. Especially content optimized for screen resolution leads to immediate shortage of space. The resolution of a projector used within an interactive whiteboard is often equal or even below the screen resolution of desktop workstations or laptops. It is necessary to minimize content like web-snippets or images. Therefore, a standard size for web-snippet viewers and for images is defined as described in section 4.1.3. This prevents the whole whiteboard from being fully covered with just a few web-snippets, or images.

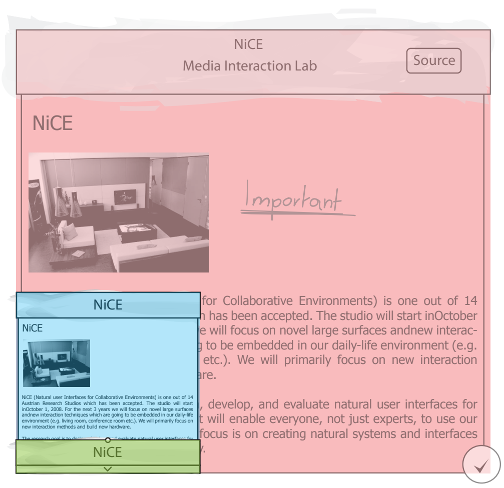


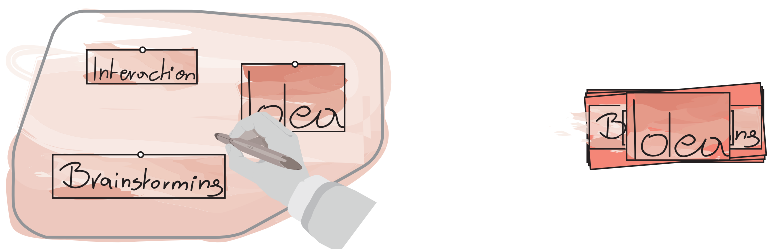
Figure 4.15: Space saved by shrinking down web-snippets. The red deposited area visualizes the original size of the snippet, as it is also displayed in edit mode. The blue area shows the standard visualization using a thumbnail, while the green area visualizes the space required when the Viewer is collapsed.

The second problem is about the placement of external content. Everything is easy, as long as the content is created directly on the screen because there is a specified place of creation. However, this place, represented by its coordinates, falls away, when the content is provided externally. Therefore, an intelligent placement algorithm that detects empty space to display is required, but even when there was enough free space to add external content all the time, such a solution leads to different problems too. In order not to lose any externally committed ideas, it is necessary to store them on a server. If the corresponding session is opened, the ideas are downloaded and displayed on the screen, but to delete them from the server automatically after the download makes no sense. If the session is not saved, the data would be lost. In summary, there is a problem of limited space and a problem of obscure data management about what data is external or local.

A basic solution to these problems is the implementation of a special bar at the top of the screen, called *Mailpipe*. All the external contributions are listed in this pipe, using small thumbnails. The user can now drag the object into the working space to have a closer look at them or to include them into the brainstorming session. However, the item is not deleted from the Mailpipe when it is used. So the Mailpipe in some respects displays all externally added ideas to the session.

4.5.3 Grouping and Stacking Items

All along automatic shrinking of handwritten content and standard use of thumbnails for images and web-snippets, groups provide another mechanism to save space. In practice, a group requires even more space than its including items (see figure 4.16(a)). However, every group can be collapsed to a stack, which reduces the required space to a single item, as illustrated in figure 4.16 (b). Later on, if the content is needed, the group can be enlarged again. The



(a) (b)

Figure 4.16: (a) A group in its ordinary shape requires plenty of space. (b) By a single tab on the group background, it is collapsed to a stack that requires much less space, when tipping on the stack again, the group is restored in its previous arrangement.

whole arrangement of all items within the group is restored. This provides an easy way to acquire additional space.

4.5.4 Instancing to Gain Space

In the first row, Instances serves another purpose than providing a virtual extension to generate more space. As described in section 4.3.4, the actual idea is to generate a copy of all the available ideas. However, the mechanism can also be used to create new pages that provide additional space, when the copied content is deleted.

Chapter 5

Implementation

This section provides an overview over the environment that was developed to realize the idea of combining verbal and electronic brainstorming. In order to do so, it describes the interfaces that enable communication between various services and the whiteboard application. The section also provides details about the general structure of the implementation. Furthermore, three mechanisms within the NiCE Brainstorming whiteboard application are presented more closely.

5.1 The NiCE Brainstorming Environment

Figure 5.1 provides an overview of the different possibilities to add content from external devices that are implemented yet. In the future, maybe other services and ways to contribute content are added. There is a couple of options how information can be contributed. Unfortunately, currently there is no option to access information from external devices.

Content Contribution per Server Socket

First of all, a regular server socket is provided by the whiteboard application that can be used to deliver information from nearby located devices, as interactive whiteboards that run a sketching application. The greatest benefit of the client server connection is the speed as it is practically instant.

Content Contribution per Mail Account

The second interface is using a regular mail account, like Gmail[1]. The use of mail to deliver information maybe seems antiquated, but it offers several advantages within this scenario. First of all, ideas can be contributed anytime without the need of running the whiteboard application all the time.

[1]http://mail.google.com

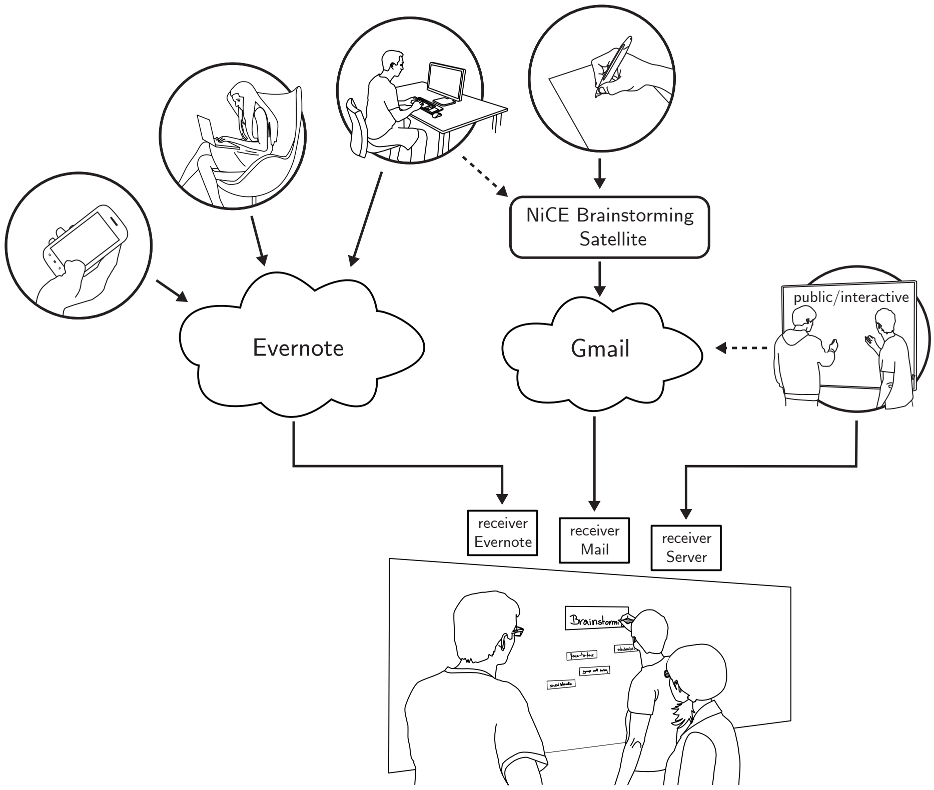


Figure 5.1: Overview over the currently implemented NiCE Brainstorming environment. Various devices can be used to contribute content by using different services. From the whiteboard application point of view there are three ways to receive content: client-server connection, e-mail, and Evernote.

Moreover, the contributions are stored in the mailbox. Therefore, they are not lost when the application is closed without saving. The only drawback, compared with the server connection is the short delay, but for most distributed and asynchronous scenarios, a delay of a few second or more is no problem. The actual delay depends on the refresh rate of the mailbox within the whiteboard application. A lower cycle then 10 seconds usually makes no sense and in practical use also a higher refresh rate is no significant problem. To receive the mails from the server, the IMAP client library InterIMAP[2] is used. Since it would be inconvenient to write a mail using a standard mail program, we developed a slim program, called *NiCE Brainstorming Satellite* that does the job. This way there is no need to worry about entering the right address and using the correct form within the mail. However, the main task of NiCE Brainstorming Satellite is to provide an access point for the

[2] http://interimap.codeplex.com/

support of paper-interaction. In the scenario framing, we assumed that most people would use paper on or nearby their desktop. Therefore, the digital pen establishes a connection to the program and streams the data to the computer, where it is stored. When a tip on the printed button indicates that the content is about to be sent, NiCE Brainstorming Satellite sends a mail to the corresponding mail-account and attaches the strokes to the mail as an ISF[3]-file. Besides forwarding digitized notes on paper, NiCE Brainstorming Satellite is also capable of sending text or locally stored images.

Content Contribution using Evernote

However, when it comes to images or even more complex content, the use of Evernote is usually the right choice. The benefit of including a service like Evernote is the support of various platforms and devices. The data is stored on the Evernote server and can be accessed by using the according API. Moreover, the whole content is provided in a single data format. It is easier to support this one format than different ones for each type of data. Each user who wants to contribute ideas has to register an Evernote account. Then the according software can be installed. There are various versions that support different devices and operating systems. In addition, Evernote can be included into every modern browser. If there is some appealing information on the web, it is enough to mark the content to be shared and press the Evernote button. The content is automatically transmitted to the personal Evernote account. Even if the browser is not supported or if it is difficult to install software on the machine, there is a way to use the service. A Bookmarklet can be used to contribute content without installation of any software and without using the clumsy web interface. Within the account, the notes are organized in multiple notebooks. These notebooks fit perfectly to the idea of multiple brainstorming sessions. For each session, an independent notebook is created to store the according content. When the whiteboard application connects to the server via the Evernote API, it gathers the submitted content from the different accounts and the corresponding notebook of the session. Alike the mail server, the application has to check for new notes periodically. Usually the refresh rate is lower here because of more complex data transfers and slower servers. Therefore, 20-30 seconds are maximum.

The three described interfaces are all the connections that are available in the current version, but due to the flexible design it is easy to connect additional services.

[3] http://msdn.microsoft.com/en-us/library/ms840393.aspx

Technology

The whiteboard application *NiCE Brainstorming* and the client application *NiCE Brainstorming Satellite* were both developed using .NET 4.0 using C# and WPF[4]. The interaction on the whiteboard, as well as the paper-based interaction, are based on the technology of Anoto[5]. Anoto pens use a tiny infrared camera to scan a dot-pattern that is printed on the according surface. Pen and computer communicate via Bluetooth, to exchange about the current position of the pen. Due to a unique ID of each pen it is possible to identify the different users within the application. This way a multi-user system can be developed.

5.2 MVVM

Within the aggregation of devices, services and applications, the design and implementation of the interactive whiteboard application obtained the highest attention. Because of the size of the application, architecture of the software was a big issue from the beginning. The overall design takes heavy use of the *Model-View-ViewModel* (MVVM) concept [30]. This design pattern aims for a strict separation between Model and View. In doing so, the View provides the visualization based on the data provided in the Model. The basic principle is that the View is unaware of the Model and vice versa the ViewModel and the Model are unaware of the View. The use of this pattern leads to a loosely coupled design, where most items or rather classes that exist within the Model, have a counterpart within the View. The proper use of the MVVM design pattern requires some mechanisms that are extremely specific to WPF. Therefore, they are mainly used in combination.

5.2.1 High Level MVVM Implementation

Since the overall idea stated in the concept is to visualize movable items on a whiteboard, it is obvious to define a Model that contains all the items and to implement their visualization within the View. Figure 5.2 provides an overview of the communication between View and Model. As the Model itself is only storage, it is administered by the ViewModel. The ViewModel provides communication between Model and View. Therefore, it contains all the business logic. Actually the coupling between the View and the ViewModel does not strictly follow the rules of the MVVM principal. While MVVM propagates that the view has to be unaware of the ViewModel, this is not the case our implementation. The reason is the highly dynamic and interactive user interface that is implemented within the View. Typically the communication in between is based on commands [23] and data binding [29], two very

[4] Windows Presentation Foundation
[5] http://www.anoto.com/

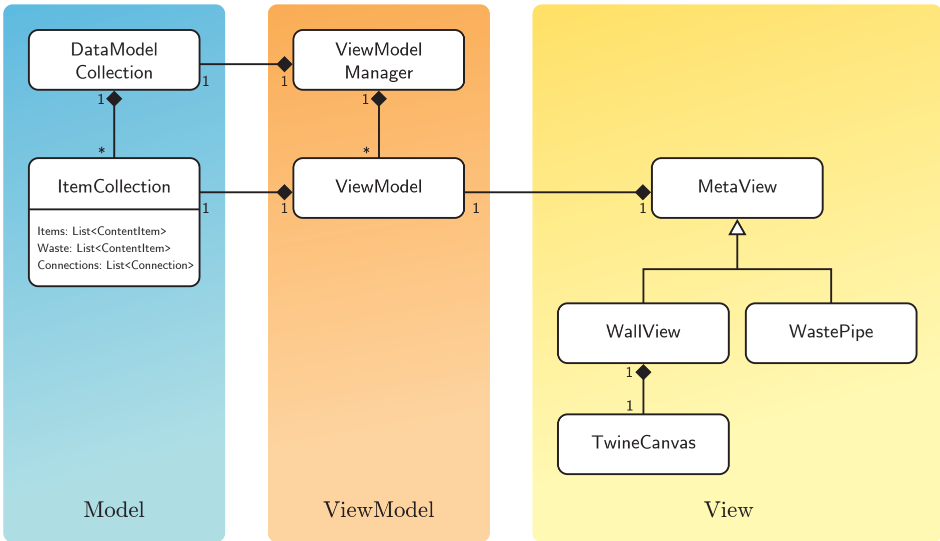


Figure 5.2: Overview over the general application structure of the NiCE Brainstorming whiteboard application. The design follows the MVVM pattern. The Model contains the pure data, while the ViewModel works on the Model and connects it to the View. The View is listening to the changes within the Model and visualizes them.

specific WPF features. Within our implementation, data binding is used extensively because it enables two properties to share the same value without writing any code to update each other. Commands, on the other hand, are difficult to use within a dynamic and object-oriented interface, where actions are not executed by pressing global buttons but by shifting around items, using controls directly on these items and performing gestures.

As seen in figure 5.2, practically there are two levels of communication. The upper level becomes necessary because of the use of Instances that virtually provide additional pages for input. Since each `ItemCollection` contains all items for one Instance, it becomes necessary to handle multiple `ItemCollections`. Within the View, `MetaView` provides a common basis for different implementation. `MetaView` contains a single `ViewModel` and this way observes the changes within the Model. As mentioned above, this approach is not strictly correct according to the MVVM paradigm. The changes within the `ViewModel` are provided to concrete implementations of `MetaView` by using events. If a new item is added, the corresponding event is thrown, and `WallView` creates a new Viewer that visualizes the item. This way, a link for new implementations of `MetaView` that provides different visual interpretation of the data model is created. To change the active Instance `ViewModel` within `MetaView` is simply exchanged. This is another advantage of the tighter coupling between View and ViewModel.

5.2.2 Low Level MVVM Implementation

Of course, the MVVM concept is not only used within high-level communication between Model and View but is also continued on lower levels. Figure 5.3 illustrates the design on the level of single items. Within the Model, each item is deduced from `ContentItem` and `BindableObject`. `BindableObject` provides methods to observe changes within the properties of each item. Each `MetaViewer` contains a single `ContentItem`. Similar to the observation of `ViewModel` by `MetaView`, each `MetaViewer` observes its corresponding `ContentItem` for any changes. The changes are passed on by using dedicated events, the classes deducted from `MetaViewer` listens to. For each type of content, there is a specific item that provides a unique interface for adequate data storage. Any item within the Model can be visualized within the View using its corresponding Viewer. This way it becomes easy to exchanges the visualizations of different types of content. However, this is a highly efficient way of communication between Model and View that also needs only loose

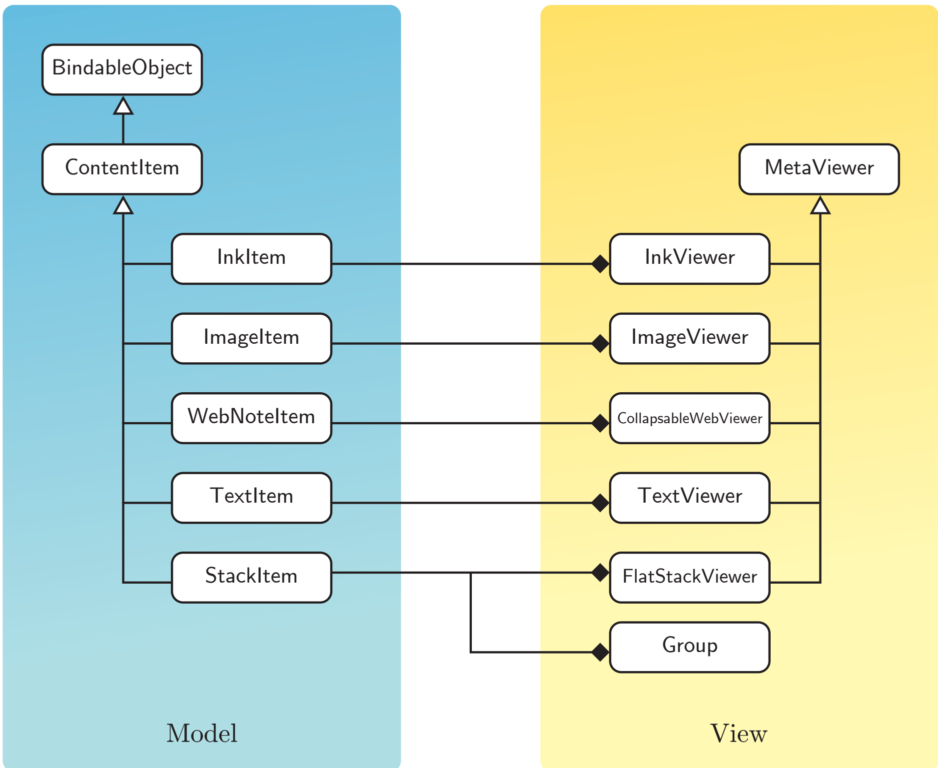


Figure 5.3: Communication between Model and View on the level of single items. Each item needs a corresponding Viewer within the View to be visualized.

coupling, strictly speaking it is not MVVM, as there is now ViewModel that separates Model and View. However, since a separation layer would only pass on properties in this case, because there is no need for any business logic on this level, we decided to do it without.

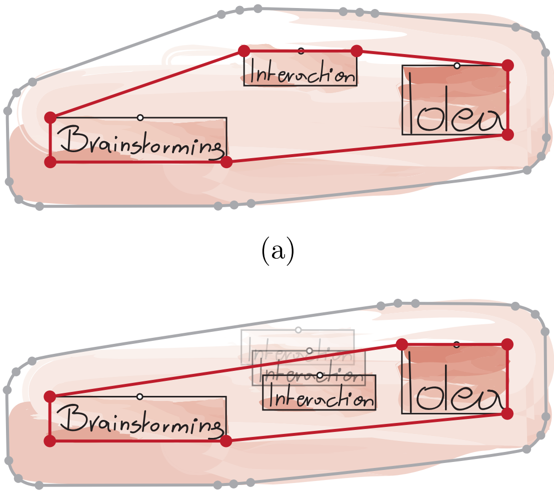
In some respect, the `InkViewer` is somewhat an exception. Ink is the only input method directly on the whiteboard, disregarding Drag & Drop. Each time a user contributes content by writing on the interactive whiteboard a new `InkItem` is created. However, besides writing down ideas, this input method is also used to perform gestures to delete content or to select multiple items. To sustain separation between visualization of handwritten content and functionality in terms of gesture recognition, `InkViewer` passes its first stroke over to `WallView` that performs the recognition. When a gesture was identified, the corresponding command is performed, and the `InkItem` and at the same time the `InkViewer`, is deleted.

Another exceptional case can be recognized within figure 5.3. A `StackItem` can be visualized in two ways. Either as a Group or collapsed to a Stack. Theoretically it would have been possible to use one and the same Viewer for both visualizations, but in order to achieve a maximum of possible interactions between the Viewers within and outside the Group, the visualizations where implemented separately. This means each `StackItem` contains a flag that specifies the current visualization. Since there is no further difference in the structure of the Model, this indicator is sufficient. In addition to flexibility in use, the MVVM concept provides a clean separation between data and visualization. This is particularly helpful when implementing save and load functionality.

5.3 Dynamic Convex Hull Calculation

As described in section 4.3.3, all items within a group are covered by a convex hull. This hull is calculated using the corner points of the items. In addition, an appropriate offset is added to enlarge the hull. The hull has to be flexible to enable rearrangement of the items within the group, but when the hull would be adapted in real time, there would be no way to rip an item out of the group. The idea is to slow down the adjustment of the hull in a way that, on the one hand, enables pulling out an item with a fast move and, on the other hand, allows comfortable rearrangement of the items within the group.

So the challenge is to slow down the adjustment of the hull. The first intuition was to slow down the adaption of the hull, when a movement of an item occurs, by interpolating the points of the hull. In fact, this implementation uses two hulls. The points of the hull from the previous visible state are matched with the points from the current hull, which represents the target, in order to find the corresponding pairs. The vector between



(a)

(b)

Figure 5.4: (a) The convex hull needs 7 points (red) to inherit all the items. (b) One item is moved and the number of points is reduced to 6. This variation in the number of points makes it difficult to implement a smoothing algorithm that works directly on the hull. The gray dots visualize the hull points after applying the offset on the hull.

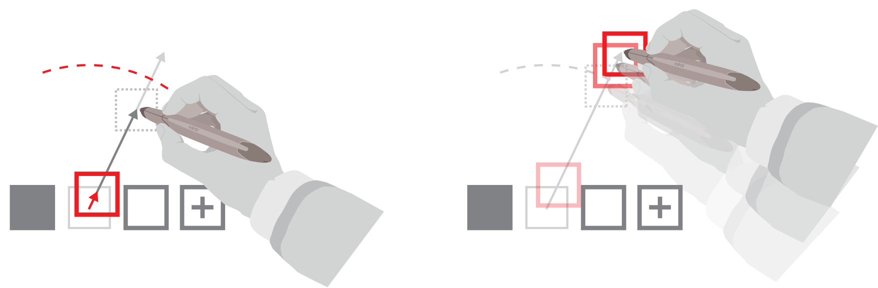
those two points is used to interpolate according to a timer. In doing so, it is necessary to use the convex hull that is calculated before the offset is applied because this reduces the number of points tremendously and this way increases the performance. However, there are still unsolved problems using this approach: There are different scenarios where the number of points within the hull changes when an item is moved (see figure 5.4). This leads to problems within the calculation and different approaches to resolve the issue result in recognizable artifacts.

Therefore, another approach was developed. Since the convex hull is calculated by using the corner coordinates of the items, it seems reasonable to interpolate these four coordinates when the item is moved. This adapted coordinates are then used for further convex hull calculation. Of course, in this scenario, the multi-user character of the application has to be considered, as multiple users are able to move different items within a single group at the same time. Thinking of the first approach that directly interpolates the hull, this would not have been necessary. In order to perform the actual calculation, a placeholder object is created when an item within the group is moved. While the visible item is transformed in the ordinary way, the placeholder object is affected by the interpolation and then used for calculating the convex hull in lieu of the actual, visible item. This way it is possible to drag the visible item out of the hull, in order to rip it out of the group.

5.4 Snatching

Snatching simulates resistance of an object or element, when it is ripped out of its ancestral position. This method is used several times within the NiCE Brainstorming whiteboard application, for instance within the Wastepipe (see section 4.2) and the Mailpipe (see section 4.5.2), but also within the visualization and control panel of the Instances, described in section 4.3.4. Due to the frequent use, the algorithm was extracted into a special class called `SnatchHelper`. This class provides two important methods, one to be called on the initial down event on the element and another one to be called continuously when the element is moved. Typically the `SnatchHelper` is used within the element that is supposed to be moved. When using a standard translation algorithm, the current position is stored as soon as a down event occurs on the element. Each time a move event occurs, the element is moved along the delta between the stored and the new position. Moreover, the stored position is updated. This process is now substituted by the `SnatchHelper`. Instead of doing the simple calculation, as in the standard translation-algorithm, the data is sent to the `SnatchHelper` that calculates the translation.

At the beginning, it is necessary to mention some structural information. In most cases when the `SnatchHelper` is used, the affected element is placed within a panel where it is fixed in its position (e.g. `Stackpanel`). In order to rip the element out of the panel, it is necessary to put it on a separate panel that allows transformation without affecting the layout of all the other elements (e.g. `Canvas`). For this purpose, the `Down(...)` method



(a) (b)

Figure 5.5: (a) The standard translate algorithm is applied virtually. The vector between the initial position and the current virtual position is shortened by applying an adjustable factor. In order to snatch the element away from its position, the barrier has to be exceeded. (b) When the barrier has been passed the element instantly jumps to the position where the last move event occurred and continues movement using the standard translation algorithm.

of the `SnatchHelper` class awaits two parameters, the position of the contact point relative to the separate panel, where the element is supposed to put onto and the position relative to the element that shall be moved. The method returns the transformation to be applied on the element to stay on its position, when it is put on the separate panel.

When the element is moved, the `Move(...)` method of the helper has to be called continuously. The actual position is calculated just alike when using the standard translation algorithm. But now additional calculations are performed to simulate an invisible barrier that prevents the element from being removed.

In order to do so, internally the standard moving-algorithm is used to calculate a virtual position of the element, as illustrated in figure 5.5 (a). This position is updated continuously. Within each cycle, a vector is calculated between the virtual position of the element and the initial position that was stored when the down event occurred on the element. This vector is shortened to a great extent by an adjustable factor and is used to reduce the effect of the translation, as long as the barrier is not passed. If the translation process is aborted before passing the barrier, the element is reset to its initial position. When the barrier has been passed, the element can be moved in a normal way (see figure 5.5 (b)). If this is the case, the `SnatchHelper` throws the so called `SnatchEvent` that indicates that the barrier has been breached. From now on, the element can be removed from its position and behaves like when using the standard translation algorithm.

5.5 Scratch Gesture

As described in section 4.2, the scratch gesture is used to remove content on multiple levels. When used within a single item in edit mode, it removes all strokes hit by the gesture. Performing the gesture in standard workspace, all items hit by the gesture are deleted, as well. The gesture can also be used to delete connections. However, connections and items cannot be removed at once; connections are prior and removed in the first place, as it is the action of less impact.

Regardless the level of interaction, the process of gesture recognizing is very much the same. For this reason, an independent mechanism was developed to recognize gestures. The interpreter is called by any item and any view that requires gesture interpretation.

There are only two, extremely basic gestures implemented within the NiCE Brainstorming whiteboard application, but those are fairly different. In contrast to the scratch gesture, encircling, to select multiple items, requires constant feedback already during the selection process. The interpretation of the scratch gesture is done, not until the tip of the pen is lifted from the surface. The above-mentioned interpreter only handles gestures that are in-

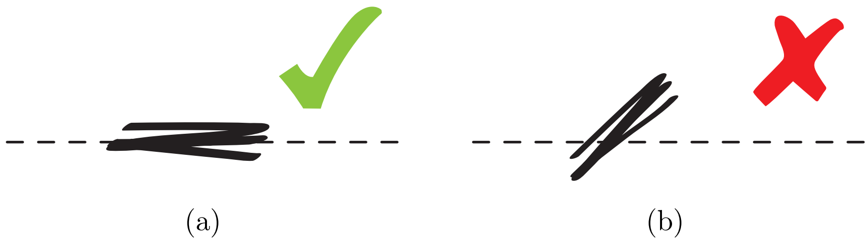


Figure 5.6: (a) The gesture is only recognized when the main direction is horizontally. (b) Otherwise the .NET GestureRecognizer does not identify the gesture as performed correctly.

terpreted after finishing and since this type only includes the scratch gesture for now, this is the only gesture implemented.

Every execution of the scratch gesture consists of two different requests. In the first place, it is checked, if the gesture is performed correctly. The second step validates, if one or multiple objects are hit. According to the level of interaction, these objects might be strokes, items or connections. If no object is hit, the scratch gesture is invalid, even though it was performed correctly.

Within the first step that recognizes if the gesture itself is performed correctly, the standard `GestureRecognizer`[6] provided in .NET is used. Among many other gestures, this `GestureRecognizer` is also capable of recognizing a gesture named *ScratchOut*. However, this gesture might seem a perfect match, testing identified shortcomings in the recognition. Figure 5.6 illustrates the problem. The .NET `GestureRecognizer` only recognizes the scratch gesture when it is performed horizontally, which is quite difficult to do on a vertical surface, as a whiteboard. Therefore, an algorithm was

[6]http://msdn.microsoft.com/en-us/library/system.windows.ink.gesturerecognizer.aspx

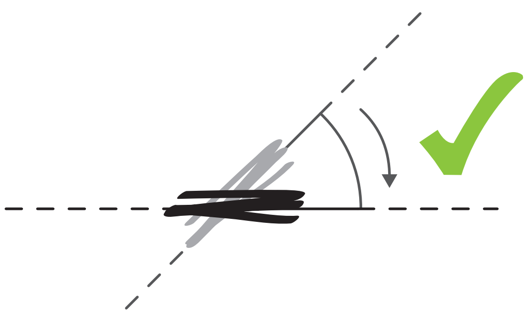


Figure 5.7: Before analysing the stroke to recognize the gesture, the main orientation is determined and the stroke is rotated into horizontal position.

implemented that prepares the submitted stroke before it is forwarded.

At first, the general orientation of stroke is determined by fitting a line into the single points of the stroke, using *Least Squares Fitting* [39]. The angle between this line and a virtual horizontal line is calculated and used to rotate the stroke into a horizontal position (see figure 5.7). Now the stroke is passed over to the `GestureRecognizer` to determine, if the gesture is successful.

In the second step, the algorithm checks, if an object is hit by the gesture. Whether an item is hit or not is easy to check. An item is hit, if at least one point of the stroke is located inside the item. Checking, if another stroke or a connection is hit, is more complicated. Basically the problem is all the same. Each stroke consists of multiple closely located points that are all connected with straight lines. The solution is to iterate over all the short lines while checking if the lines are crossing the other stroke or the connection. If the object to be compared is a stroke too, it is treated in the same way. The two strokes are split into short lines, and each pair of lines is checked for crossing. Since a connection is only one straight line, it is even easier here. If any crossing occurs, the object is regarded as hit and is deleted in consequence.

Chapter 6

Discussion

6.1 Evaluation

Measuring the success of a brainstorming process is a difficult thing to do. Although there are plenty of quantitative studies that measure productivity to compare different methods [25, 28], recent research [7] and the success of verbal brainstorming indicate that productivity is not necessarily a profound value of measurement. Other studies [26] also observe the originality and the feasibility of the produced ideas. However, in some respect these values are subjective and it turned out that participants are extremely weak in estimate their performance, following criteria of originality and feasibility. Hilliges et al.[16] compared an interactive system to its paper-based equivalent using quantitative and qualitative methods. In the case of the system we presented, this is very difficult to do, since it is hard to simulate the entire, distributed system, including external devices by using a paper-based equivalent. Therefore, we restricted ourselves to informal observations of external people using the system and to our own experiences while using NiCE Brainstorming within our own brainstorming sessions.

In doing so, we found out about the strengths and weaknesses of the system, especially of the interactive whiteboard application. When using it, we were very pleased about the fluid interaction when creating content. As mentioned in section 4.1.1, the dynamic adjustment of the Post-its was a matter of discussion. Fortunately, the concern that the mechanism would be abused, and users would tend to write everything on a giant Post-it, has not come true. On the contrary, the use of Post-its for visualization turned out to be a good choice, since it encourages keeping written phrases short. Overall, the dynamic adjustment works remarkably well from the perspective of interaction, but it leads to more technical effort. It is not a matter of the implementation that calculates the adaption. However, handling items of various size and ratio complicates the implementation of other features.

Furthermore, connections and groups emerged to be easy to use and turned out as particularly helpful to structure content and this way, to provide additional value to the information on the interactive whiteboard.

Despite the well-working basic functionality like fluid content creation and powerful structuring opportunities, there are still several issues to be improved. The application uses three global interface elements (Wastepipe, Mailpipe, Instance visualization). The concept works fine using a single-screen setup with moderate dimensions, but when it comes to the use of multi-screen setups this interface concept comes to its limits.

6.1.1 Wastepipe

Using the Wastepipe becomes increasingly difficult when multiple users are working on a large scale interactive whiteboard, as deleted items are stored continuously from left to right within the Wastepipe. The need to change the position to restore an accidentally removed item may disrupt other users within their considerations. Placing the deleted items right beneath their original position would be a possible solution. However, it does not change another problem. As described in section 4.2, the Wastepipe is a utility to restore accidentally deleted objects. When using the application, only a small number of deleted items are effectively restored. On the one hand, this indicates that the scribble gesture for removing items works exceptionally well, since few items are deleted by accident, but on the other hand, it challenges the concept of the Wastepipe that it is trashed with fractures of ideas or even with accidentally created strokes that will never be restored. Therefore, the whole concept needs further investigation, to substantiate the needs within the system to restore unintentionally deleted items.

6.1.2 Instances

Though the visualization of the Instances is a global interface element too, it is a minor problem on large whiteboards due to its infrequent use. Moreover, creating a new Instance on the whiteboard is a global task. Therefore, all the participants within the brainstorming session are affected in any case. Because of this, the disruption when somebody is moving toward the border of the board is not that severe. The more interesting question about Instances is related to their intended purpose of providing a way to structure equal content in a different way. It is hard to tell if this approach generates any additional value, or if it would be more useful to provide a way to add additional blank pages, in contrast to copy all items. Possibly there are other methods that provide the intended functionality in a better way.

6.1.3 Mailpipe

The Mailpipe provides a straightforward solution to the very complex topic of importing external content, but of course the solution it is not without its drawbacks. The similarity of Mailpipe and Wastepipe is distinctive, whereupon it is important to mention that the placement on the bottom respectively on the top is no coincidence but follows a the logical principle. While items on the top are perfectly visible to all participants, the bottom is out of focus most of the time. Since the Mailpipe, in contrast to the Wastepipe, is less integrated into the continuous work of multiple users on the whiteboard, the implementation as a global interface element is less a problem.

The troubles of placing external contents are described in section 4.1.3. The Mailpipe is resolving issues of placement and session dealing at the same time, but it does not provide any additional information about who sent the content, what device was used to send it, or when it was sent. While it is pretty obvious that information about what device was used to submit an idea is not substantial, and it is hard to imagine that the time of contribution is very informative, the question if it is valuable to know the contributor, is far more difficult. Although anonymity is one of the big advantages in electronic brainstorming because *evaluation apprehension* is avoided, it is not assured that it is helpful within small groups (up to 6 people). This assumption is affirmed by the findings of Liikkanen et al.[20]. Anonymity was a design goal when developing the brainstorming tool Presemo Brainstormer. However, the evaluation in groups of 5-6 evidenced that that people tried to resolve the anonymous identities of others. This indicates that, within small groups, where people know each other very well, it would be easier to unveil identities because some contributions might be assigned to its author anyway. Therefore, it would be helpful to unveil the creator, to avoid people exchanging and guessing about authorship. Theoretically it should also provide a benefit in terms of *social loafing*, but at the same time result in a drawback respective *evaluation apprehension*, although this effect is possibly reduced a little within a distributed and individual idea generation phase. Anyway, unveiling personal information can also lead into the wrong direction, since ideally the focus should stay on the ideas and not on the author. Moreover, the permanent presence of this information during face-to-face brainstorming might lead to a more subjective view on some contributions.

6.1.4 Layout

An issue connected closely to the placement of external content, is functionality that helps with arranging items. The idea is not about any guides or a hidden grid that allows content snapping. Since content can have various size, this would provide only little value. PostBrainstorm [14] provides the ability to create lists from selected items. In some respect, the mechanism is similar to the definition of groups within NiCE Brainstorming, although it serves another purpose. However, we believe that a functional layout mechanism should provide a more global concept.

For this reason, a prototyped implementation of layout functionality was

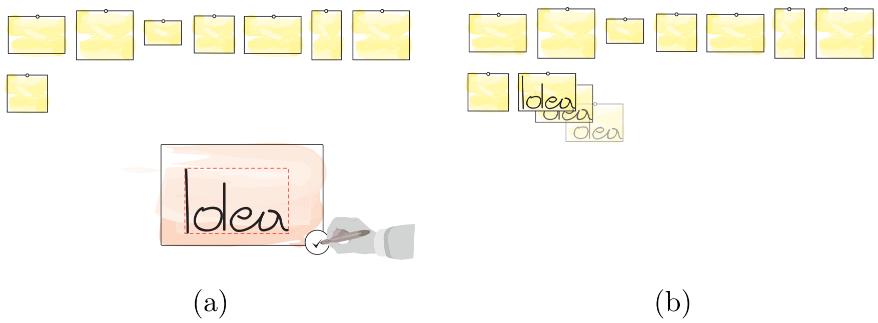


Figure 6.1: Mode one: (a) Immediately after a new item is created and finished, (b) it is automatically moved to the top to clear the space in the preferred height of writing.

included into NiCE Brainstorming. It provides two modes: The first one influences the layout immediately when the content is created (see figure 6.1), while the second one arranges all existing items as soon as a corresponding command is executed (see figure 6.2). In each case, the items are arranged in a grid pattern, whereupon starting point and direction are adjustable.

Especially when using the first mode, content placement on the top is reasonable. In contrast to a brainstorming session on a whiteboard, where most people start to write on the upper left to use the available space in an efficient way, we found out that people using NiCE Brainstorming have a preferred height to write. Soon the area within this height is clustered with items, but as it is easy to move items away, this is no problem. So the idea was to simplify the process by providing an automatic mechanism that moves the items away as soon as they are written. It would be ideal to move them to the top, so that they are easy to spot.

The second mode uses a similar mechanism. However, it does not immediately move the items after its creation, but it waits until the user executes the action. The idea is to provide a structured overview over all items after the phase of idea generation, when the participants are about to evaluate and select. Thinking further, this provides the basis for additional features that support the selection and evaluation process. For instance, it would be imaginable to perform a voting where the favorite ideas are pointed out by each participant, drawing a stroke or dot on the item containing the idea. At the end, the system is capable of ordering the items by the number of votes they obtained.

Despite promising approaches, the system shows its drawbacks within practical use. A simple grid to arrange the items leads to problems when items are very different in size. This fact complicates the implementation because a flexible system is needed that is able to cope with different sized items and arranges them in an adequate way. However, the critical issue is

(a) (b)

Figure 6.2: Mode two: All the items are arranged automatically when the according command is executed.

not the pure arrangement, but more the integration of the layout mechanism within the overall brainstorming concept. In section 3.2.1, the decision to implement a system that allows flexible use and supports different varieties of use is argued. Concerning the layout mechanism, the freedom of the process leads to several problems. Therefore, if the first mode is activated and the items are automatically moved to the top (see figure 6.1), these items can be removed or moved anywhere else by users. In that case, various possibilities are imaginable. The automatic arranging-mechanism could be disabled if the user interferes with it. The placement algorithm could also be adapted to arrange new items into the arising holes or shift other items to close them. It is also hard to tell what is supposed to happen when there are already existing items within the application, and the automatic, arranging mode is activated. To rearrange them would be highly problematic if they are already structured within groups or connected with each other by the user. Otherwise, it would be hard to arrange new items automatically when existing items block the space, where new ones are supposed to be placed. Using the second mode (see figure 6.2) is even more difficult because arrangements by the user are ripped apart, when the algorithm is not smart enough to gather this information out of the current placement that may be user intended or not. In conclusion, the implementation of a system that supports the user by automatically arranging and moving items would be a substantial benefit for the application. However, it is a difficult challenge to use machinable mechanisms within an application that supports a very loose process.

6.1.5 Finite Space

Just alike each physical whiteboard, NiCE Brainstorming provides only finite space. In section 4.5, several mechanisms are presented to overcome this problem. However, finite space is not only a problem when handling plenty of content, but also when using an object-oriented interface. Especially when items are within edit mode, and they are enlarged, it can easily happen that single controls are suddenly outside the displayed area, and therefore out of access. Moreover, at this time it is impossible to move them. Especially web-snippets can lead to urgent space problems when they are opened to full size. As a result, the controls are never the only way to change back from edit mode. A simple tap on any other item or empty space resets the item from edit mode and makes it movable again. Nevertheless, it would be helpful, if a mechanism were implemented that observes the placement of controls and adapts them if necessary. In doing so, there is also the issue of shortage in space when writing down an idea near to the right border. Although similar restrictions exist on each traditional whiteboard, it would be helpful if an according mechanism could solve this problem too, as it is intensified by the reduction of the item size of handwritten content and this reduction leads to wrong estimations about the space, text fragments needs.

6.2 Future Work

In the meantime, NiCE Brainstorming has become a helpful tool we like to use in our own brainstorming sessions. Even so, there are still several issues to be improved, and numerous features to be added. Primarily the problems mentioned in section 6.1 have to be undertaken a critical investigation to improve the interaction within the main application on the interactive whiteboard. Another focus is the completion of the technical requirements to make the whole brainstorming process work, as described in figure 3.2. In particular, this includes the implementation of a return channel that enables external devices to access all the contributed ideas. This way the asynchronous and distributed use of the whole system is strengthened.

Another issue is the support of other phases than the actual idea generation. As mentioned in section 3.2.1, brainstorming is not only about creating lots of ideas but also about the initial problem framing and of course the final solution finding. The general idea of face-to-face brainstorming supports a collaborative process of problem framing also selection and evaluation. Although there is an easy way to move items and powerful methods to structure, in order to support the evaluation and selection process, there is still no dedicated help for framing the problem that is provided by the system. Though it is a challenge, it would be an enormous benefit to find some methods to support this process. Moreover, additional help for evaluating ideas would also be welcome.

As described in section 1.2, the goal of NiCE Brainstorming is to offers a digital working environment similar to pen and paper in the analog world. While the advantages of digital communication methods should be offered and integrated into the process, it was important to provide the freedom to use the application the way it fits best to their familiar way of brainstorming. However, providing the user with the ability to choose also includes disadvantages. On the one hand, the process can be adapted to ones needs, but on the other hand, it requires knowledge about the subject to use it in a meaningful way. In other words, the current concept provides a platform and an environment to perform brainstorming successfully, but it does not provide any guidance or help about how it works best. The idea is not to restrict the user and dismiss the idea of providing freedom regarding the process, but to offer help for common problems. In concrete terms, this means to provide some kind of storyboard within the application to support the use of dedicated creative techniques that provide good prospects of success within a special scenario. Due to versatile possibilities and devices that are included into the brainstorming environment, the implementation of diverse collocated and distributed techniques is feasible.

For Instance, it would be possible to create paper sheets with a special design to provide a template for the 6-3-5 technique [27]. The technique is done the usual way passing round sheets of paper, but at the end, it is not necessary to copy the ideas manually to the public interactive whiteboard for the evaluation and selection process.

Another technique that provides more room for improvement through external digital devices is Collective Notebook [32]. The basic idea is to carry a notebook for a couple of days, to capture the ideas that come up to one's mind. The notebook can be exchanged for a Smartphone. This way not only the stuff to be carried around can be reduced, but also the tools within the phone, as the camera or web-access can be used to record and submit ideas. Moreover, if reasonable, the ideas already generated by others can be accessed and used to build own ideas on top. Finally, it is comfortable to use automatic data transfer to the interactive whiteboard instead of copying all data manually.

Alongside these additional features and improvements, the objective evaluation of the system is a crucial issue. As mentioned in section 6.1, the free concept exacerbates the comparison of NiCE Brainstorming to other systems. Maybe the implementation of various dedicated, creative techniques helps to evaluate and compare at least single aspects of the system.

Chapter 7

Conclusions

NiCE Brainstorming is a digital brainstorming environment. Its core is an interactive whiteboard application that enables the use of verbal respectively face-to-face brainstorming methods with digital tools. Thereby users have the freedom to carry on doing brainstorming in their favorite and familiar way, since the application was designed on a very generic level. The application enables fast idea generation and fluid methods to transform and structure. Beside the support of well-known verbal brainstorming, NiCE Brainstorming also contains elements of electronic brainstorming and encourages their use by a smooth connection between the two techniques. This implies a fluid change between the alternatives, without the need to manually copying content. Various external devices are supported to contribute ideas in a distributed and asynchronous way. This does not only include desktop-computers, laptops or Smartphones but also traditional mediums as paper or whiteboards when their content is captured digitally. Moreover, ideas can be submitted from any browser-equipped device because NiCE Brainstorming supports the web-service Evernote that allows the storage and management of various types of content. This way, a wide range of devices is supported and can be used for idea contribution.

It is difficult to evaluate a system that provides versatile ways of use as a whole, but it is also a challenge to evaluate brainstorming processes in general. Since the main motivation of designing the presented system is based on research that doubts the significance of productivity as only valid measurement in classical brainstorming literature [7, 26, 34], it is not reasonable to use the same method of measurement to verify the result. However, especially the whiteboard application showed promising results during first informal testing, and we increasingly use it within our own brainstorming sessions. Even though we became aware of different problems, the overall result is decidedly positive. During the evaluation process in preparation for this Thesis, we found various possibilities to improve it even further, and we hope to include them into the application soon.

Appendix A

DVD Content

Thesis

Pfad: /

Florian Perteneder_Thesis.pdf Masterthesis

Miscellaneous

Pfad: /documents

images/ . . . . images and illustrations
online-sources/ . . . . . copys of online literature

Bibliography

[1] Barker, R.: *The Art of Brainstorming.* BusinessWeek, August 2002. http://www.businessweek.com/magazine/content/02_34/b3796663.htm.

[2] Bastéa-Forte, M. and C. Yen: *Encouraging contribution to shared sketches in brainstorming meetings.* In *CHI '07 extended abstracts on Human factors in computing systems*, pp. 2267–2272. ACM, 2007.

[3] Beyer, H. and K. Holtzblatt: *Contextual design.* interactions, 6:32–42, January 1999.

[4] Brandl, P., C. Richter, and M. Haller: *NiCEBook - Supporting Natural Note Taking.* In *Proceedings of the 28th international conference on Human factors in computing systems*, pp. 599–608. ACM, 2010.

[5] Buxton, W.: *Chunking and Phrasing and the Design of Human-Computer Dialogues.* In *Proceedings of the IFIP World Computer Congress*, pp. 475–480. North Holland Publishers, 1986.

[6] Collins, D.: *Designing Object-Oriented User Interfaces.* Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1994.

[7] Dennis, A.R. and B. Reinicke: *Beta Versus VHS and the Acceptance of Electronic Brainstorming Technology.* MIS Quarterly, 28(1):1–20, 2004.

[8] Dennis, A.R. and J.S. Valacich: *Research Note: Electronic Brainstorming: Illusions and Patterns of Productivity.* Infomation Systems Research, 10:375–377, April 1999.

[9] Dix, A.J., J. Finlay, and G.D. Abowd: *Human-computer interaction.* Pearson Prentice-Hall, New York, NY, USA, 3rd ed., 2004.

[10] Gallupe, L.M. Bastianutti, and W.H. Cooper: *Unblocking Brainstorms.* Journal of Applied Psychology, 76(1):137–142, Feb. 1991.

[11] Gallupe, R.B. and W.H. Cooper: *Brainstorming Electronically.* Sloan Management Review, 35(1):27–36, Fall 1993.

[12] Geschka, H., G.R. Schaude, and H. Schlicksupp: *Modern techniques for solving problems.* Chemical Engineering, pp. 91–97, August 1973.

[13] Guimbretière, F.: *Fluid interaction for high resolution wall-size displays.* PhD thesis, Stanford University, Stanford, CA, USA, January 2002.

[14] Guimbretière, F., M. Stone, and T. Winograd: *Fluid interaction with high-resolution wall-size displays.* In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 21–30. ACM, 2001.

[15] Guimbretière, F. and T. Winograd: *FlowMenu: combining command, text, and data entry.* In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pp. 213–216. ACM, 2000.

[16] Hilliges, O., L. Terrenghi, S. Boring, D. Kim, H. Richter, and A. Butz: *Designing for collaborative creative problem solving.* In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pp. 137–146. ACM, 2007.

[17] Klemmer, S.R., M.W. Newman, R. Farrell, M. Bilezikjian, and J.A. Landay: *The designers' outpost: a tangible interface for collaborative web site.* In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 1–10. ACM, 2001.

[18] Leitner, J., C. Rendl, F. Perteneder, A. Gokcezade, T. Seifried, M. Haller, R. Zeleznik, and A. Bragdon: *NiCE formula editor.* In *SIGGRAPH '10: ACM SIGGRAPH 2010 Talks*, p. 1. ACM, 2010.

[19] Liao, C., F. Guimbretière, K. Hinckley, and J. Hollan: *Papiercraft: A gesture-based command system for interactive paper.* ACM Transactions on Computer-Human Interaction, 14:18:1–18:27, January 2008.

[20] Liikkanen, L.A., K. Kuikkaniemi, P. Lievonen, and P. Ojala: *Next step in electronic brainstorming: collaborative creativity with the web.* In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pp. 2029–2034. ACM, 2011.

[21] Mynatt, E.D., T. Igarashi, W.K. Edwards, and A. LaMarca: *Flatland: new dimensions in office whiteboards.* In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 346–353. ACM, 1999.

[22] Nijstad, B.A., W. Stroebe, and H.F.M. Lodewijkx: *The illusion of group productivity: a reduction of failures explanation.* European Journal of Social Psychology, 36(1):31–48, 2006.

[23] Noyes, B.: *Understanding Routed Events and Commands In WPF.* MSDN Magazine, September:87–95, 2008. http://msdn.microsoft.com/en-us/magazine/cc785480.aspx.

[24] Osborn, A.F.: *Applied Imagination : principles and procedures of creative thinking.* Scribner, New York, NY, USA, 3rd ed., 1979.

[25] Paulus, P.B. and V.R. Brown: *Toward More Creative and Innovative Group Idea Generation: A Cognitive-Social-Motivational Perspective of Brainstorming.* Social and Personality Psychology Compass, 1(1):248–265, 2007.

[26] Rietzschel, E.F., B.A. Nijstad, and W. Stroebe: *Productivity is not enough: A comparison of interactive and nominal brainstorming groups on idea generation and selection.* Journal of Experimental Social Psychology, 42(2):244–251, 2006.

[27] Rohrbach, B.: *Kreativ nach Regeln - Methode 635, eine neue Technik zum Lösen von Problemen.* Absatzwirtschaft, 12:73–75, October 1969.

[28] Roy, M.C., S. Gauvin, and M. Limayem: *Electronic Group Brainstorming: The Role of Feedback on Productivity.* Small Group Research, 27(2):215–247, 1996.

[29] Smith, J.: *Customize Data Display with Data Binding and WPF.* MSDN Magazine, July:78–88, 2008. http://msdn.microsoft.com/en-us/magazine/cc700358.aspx.

[30] Smith, J.: *WPF Apps with the Model-View-ViewModel Design Pattern.* MSDN Magazine, February:72–81, 2009. http://msdn.microsoft.com/en-us/magazine/dd419663.aspx.

[31] Steimle, J.: *Designing pen-and-paper user interfaces for interaction with documents.* In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pp. 197–204. ACM, 2009.

[32] Steiner, G.: *Kreativitätsmanagement: Durch Kreativität zur Innovation*, ch. 6, pp. 267–326. Facultas Verlags- und Buchhandels AG, Wien, Ö, 2nd ed., 2003.

[33] Surowiecki, J.: *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations.* Doubleday, New York, NY, USA, May 2004.

[34] Sutton, R.I. and A. Hargadon: *Brainstorming Groups in Context: Effectiveness in a Product Design Firm.* Administrative Science Quarterly, 41:685–718, December 1996, ISSN 0742-1222.

[35] Taylor, D.W., P.C. Berry, and C.H. Block: *Does Group Participation When Using Brainstorming Facilitate or Inhibit Creative Thinking?* Administrative Science Quarterly, 3:23–47, June 1958.

[36] Tesler, L.: *Object-oriented user interfaces and object-oriented languages (Keynote Address).* In *Proceedings of the 1983 ACM SIGSMALL symposium on Personal and small computers*, pp. 3–5. ACM, 1983.

[37] Van De Ven, A. and A.L. Delbecq: *Nominal versus interacting group processes for committee decision-making effectiveness.* The Academy of Management Journal, 14:203–212, June 1971.

[38] Watanabe, N., M. Washida, and T. Igarashi: *Bubble clusters: an interface for manipulating spatial aggregation of graphical objects.* In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 173–182. ACM, 2007.

[39] Weisstein, E.W.: *Least Squares Fitting.* MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/LeastSquaresFitting.html.

[40] Winograd, T. and F. Guimbretière: *Visual instruments for an interactive mural.* In *CHI '99 extended abstracts on Human factors in computing systems*, pp. 234–235. ACM, 1999.

[41] Yeh, R., C. Liao, S. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke: *ButterflyNet: a mobile capture and access system for field biology research.* In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 571–580. ACM, 2006.

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —

Breite = 100 mm
Höhe = 50 mm

— Diese Seite nach dem Druck entfernen! —