

Kollaborative Arbeitsprozesse mit vernetzten multimedialen Whiteboards

CHRISTIAN SCHAFLEITNER

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

DIGITALE MEDIEN

in Hagenberg

im Juni 2008

© Copyright 2008 Christian Schafleitner

Alle Rechte vorbehalten

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 26. Juni 2008

Christian Schafleitner

Inhaltsverzeichnis

Erklärung	iii
Vorwort	vii
Kurzfassung	viii
Abstract	ix
1 Einleitung	1
1.1 Motivation	1
1.1.1 Gemeinsames Arbeiten auf Tafeln	1
1.1.2 Verteiltes Arbeiten	2
1.2 Ausgangsszenarien und Zielsetzung	3
1.3 Gliederung der Arbeit	3
2 Visionen für das gemeinsame Arbeiten in der Zukunft	4
2.1 Digitale Wände im Einsatz	5
2.2 Vernetzte Welten	7
3 Stand der Technik und verwandte Arbeiten	9
3.1 Verteiltes kollaboratives Arbeiten	9
3.1.1 Erste Videokonferenzsysteme	9
3.1.2 <i>Chats, Instant Messaging</i> und <i>VoIP</i>	10
3.1.3 Online-Textverarbeitung und <i>Desktop Sharing</i>	12
3.2 Digitale Arbeitsflächen	13
3.2.1 Interaktiver Tisch: <i>Microsoft Surface</i>	14
3.2.2 Digitale Schultafel: <i>eChalk</i>	15
3.2.3 Digitales Whiteboard: <i>Intoi</i>	16
4 Entwurf und Konzept	20
4.1 Kollaborative Arbeitsprozesse	20
4.2 Anforderungen an den Prototypen	21
4.2.1 Natürliche Arbeitsweise	22
4.2.2 Anmeldung	22

4.2.3	Sessions und Authentifizierung	22
4.2.4	Kollaborative Funktionalitäten	24
5	Technologien	27
5.1	Interaktives Whiteboard	27
5.1.1	<i>Anoto</i> Technologie	28
5.1.2	Rückprojektionsaufbau	29
5.2	Softwaretechnologien	31
5.2.1	<i>.NET Framework 3.0</i>	31
5.2.2	<i>Windows Presentation Foundation</i>	32
5.2.3	Verwaltung mehrerer Eingabegeräte	34
5.3	Netzwerkprogrammierung	35
5.3.1	<i>Transmission Control Protocol (TCP)</i>	35
6	Implementierung	37
6.1	Anwendungsarchitektur	37
6.1.1	Client/Server Architektur	38
6.1.2	<i>Model-View-Controller</i> Architektur	40
6.1.3	Kommandos und Ereignisse	40
6.2	Datenmodell	42
6.2.1	Identifizierbare Objekte	43
6.2.2	Ereignisse bei einer Modell-Änderung	44
6.3	Netzwerkprotokoll und Serveranwendung	44
6.3.1	Wahl des Übertragungsprotokolls	45
6.3.2	Whiteboard Server	45
6.3.3	Nachrichtensystem	47
6.3.4	Dokumentenaustausch	50
6.4	Benutzeroberfläche	50
6.4.1	Grenzenlose Arbeitsfläche	51
6.4.2	Elemente auf der Arbeitsfläche	52
6.4.3	Menükonzept	54
6.4.4	Eingabedialoge und Handschriftenerkennung	55
6.4.5	Kollaborative Funktionalitäten	56
7	Evaluierung und Diskussion	58
7.1	Informeller Benutzertest	58
7.1.1	Testteilnehmer	58
7.1.2	Testaufbau	58
7.1.3	Testsznarien	59
7.1.4	Beobachtungen und Fragebogen	60
7.2	Ergebnisse und Erkenntnisse	61
7.2.1	Meinungen, Probleme und Ideen	61
7.2.2	Leistungsevaluierung	64

Inhaltsverzeichnis	vi
8 Zusammenfassung	66
8.1 Ausblick und Wünsche	67
8.2 Resümee	67
A Inhalt der CD-ROM	68
A.1 Masterarbeit	68
A.2 Prototypische Anwendung	68
Literaturverzeichnis	70

Vorwort

Die letzten beiden Jahre meiner Studienzzeit vergingen wie im Fluge, das lag vor allem an den interessanten Projekten an denen ich im Studiengang „Digitale Medien“ arbeiten durfte. Aus diesen möchte ich sicherlich das Projekt *Intoi*, ein digitales Whiteboard, herausheben, welches gewissermaßen als Grundlage für diese Masterarbeit diente. An dieser Stelle möchte ich mich bei meinen Team-Kollegen und Freunden für die schöne gemeinsame Arbeit bedanken!

Durch Präsentationen von *Intoi* bei Firmen wurden wir immer wieder auf die Notwendigkeit der Vernetzung derartiger Whiteboards aufmerksam gemacht, was mir sodann den Anstoß für dieses Projekt und die abschließende Arbeit gab.

Danken möchten ich auch meinem Betreuer Mag. Volker Christian, welcher mir vor allem bei der Projektentstehungsphase mit wertvollen Tipps und Ideen, sowie bei der ein oder anderen Schreibblockade weitergeholfen hat. Ein besonderer Dank gilt auch allen Korrektur-Lesern dieser Arbeit, die somit wesentlich die Qualität verbessert haben.

Zu guter Letzt bleibt mir nur mehr bei meinen Eltern „Danke“ zu sagen, die mir dieses Studium erst ermöglicht und mich dabei immer unterstützt haben.

Kurzfassung

Verteiltes Arbeiten ist durch eine neue Flexibilität von Firmen und deren Mitarbeitern eine dringende Notwendigkeit geworden – vor allem die Kommunikation zwischen mehreren Personen spielt dabei eine wichtige Rolle. Das Internet bietet bereits Anwendungen, die dies ermöglichen. Besonders bei Besprechungen oder kreativen Aufgaben reicht allerdings eine Audio- oder Videoverbindung meist nicht aus.

Digitale Whiteboards lösen langsam aber stetig ihre analogen Vorgänger ab und bieten neue Möglichkeiten mit einem Computersystem zu interagieren. Anstatt mit Maus und Tastatur können mit Hilfe digitaler Stifte die Vorteile analoger Handschrift mit multimedialen Inhalten verbunden werden.

Diese Masterarbeit behandelt die Verknüpfung digitaler Whiteboards mit einer neuartigen Anwendung, die einen gemeinsamen Arbeitsprozess an verschiedenen Orten ermöglicht. Neben der technischen Realisierung einer prototypischen Server- und Clientanwendung wird auch auf deren kollaborative Funktionalität eingegangen. Die vorgestellte Anwendung kann sowohl mit digitalen Stiften als auch mit Maus und Tastatur auf unterschiedlichen Geräten gesteuert werden. Die Möglichkeit der digitalen Archivierung und der Erstellung automatisierter Protokolle gestaltet die in dieser Arbeit beschriebene Anwendung für den Benutzer attraktiver.

Eine Evaluierung zeigt, dass mit Hilfe neuer Interaktionskonzepte und Funktionalität ein gemeinsames Arbeiten von Anwendern, im Vergleich zu bereits vorhandenen Desktop-Anwendungen, eher akzeptiert wird. Ein Problem, das auch von dieser Anwendung nur schwer zu lösen ist, ist eine zwischenmenschliche Kommunikation auf eine adäquate Weise zu ermöglichen.

Abstract

Distributed work has been more and more relevant through the increasing flexibility of companies and people. Moreoften the collaboration of different people on various places is necessary. The internet already offers applications to facilitate collaborative work. However, in meetings and creative works an audio- or video connection is insufficient.

Digital whiteboards are replacing their analogous predecessor and offer new possibilities to interact with a computer. Instead of mice and keyboards, digital pens are used to combine handwritten material with multimedia-based content.

The combination of a digital whiteboard with a novel application, which allows collaborated work on widespread locations, has been discussed in this Master's thesis. Besides the technical realization of a server and client prototype, features for shared scenarios are presented. The application can be operated with digital pens as well as with mice and keyboards on different devices. The possibilities to archive digital content centrally and create automated journals should encourage people to use this software.

The evaluation of the prototype shows that new user interaction concepts and functionality are rather accepted than well-known solutions. Though it is still hardly possible to allow interpersonal communication through a software application.

Kapitel 1

Einleitung

1.1 Motivation

Die Notwendigkeit digitale Whiteboards zu kollaborativen verteilten Systemen auszubauen, beruht auf unterschiedlichen Grundlagen und Gegebenheiten. Diese sollen in dieser Einleitung kurz erläutert werden.

1.1.1 Gemeinsames Arbeiten auf Tafeln

Ob in der Schule oder in Besprechungen, bei der Ausbildung und beim Vermitteln von Wissen, sowie bei kollaborativen Arbeiten spielen beschreibbare Wände oder Tafeln meist eine zentrale Rolle. Obwohl Projektoren und moderne Präsentationstechniken im Vormarsch sind, haben sich die altbewährte Schultafel in Klassenzimmern und Whiteboards in Besprechungsräumen einen fixen Platz geschaffen.

Bei kollaborativen Arbeiten, seien es Gruppendiskussionen oder Ideenfindungsprozessen, ist es notwendig allen Teilnehmern die gesamten Informationen zugänglich zu machen um dadurch neue Inspirationen zu ermöglichen. Dazu eignen sich z. B. Flipcharts hervorragend, da diese von jedem einfach beschrieben werden können und für alle Personen im Besprechungsraum sichtbar sind (vgl. Abb. 1.1). Besonders bei kreativen Prozessen kann mit Hilfe von Skizzen, Pfeilen und Anmerkungen der Denkprozess strukturiert und beschleunigt werden. Dennoch sind analoge Flipcharts oder Whiteboards in ihrer Funktionalität beschränkt, daher wird versucht diese durch digitale Technik zu ersetzen.

Digitale Whiteboards

Was vor ein paar Jahren noch als *Science-Fiction* betrachtet wurde, wird nach und nach Wirklichkeit. Berührungssensitive Oberflächen und moderne Eingabegeräte erlauben uns auf eine neue Art und Weise mit Computern



Abbildung 1.1: Bei Besprechungen ist neben einer Projektion meist immer ein Whiteboard oder Flipchart im Einsatz.

zu interagieren. Besonders der Markt digitaler Whiteboards ist einer der zukunftsträchtigsten Technologien. Mit Hilfe digitaler Technik können multimediale Inhalte ebenso präsentiert und mit analoger Handschrift kombiniert werden.

Des Weiteren ist es auch entscheidend, dass digitale Technik immer mehr auch von älteren Personen oder Führungskräften akzeptiert wird, welche an sich nichts mit Informationstechnologie zu tun haben wollen. Nicht nur, dass man heutzutage mit dieser modernen Technik, vor allem dem Internet, aufwächst, auch immer einfacher zu bedienende Programme tragen dazu bei. Besonders bei Applikationen des täglichen Gebrauchs ist es wichtig, dass diese gerne und einfach bedient werden können. Digitale Whiteboards bieten hier eine besondere Möglichkeit neue Interaktionskonzepte zu realisieren.

1.1.2 Verteiltes Arbeiten

Durch die Globalisierung sind nicht nur politische und wirtschaftliche Grenzen verschwunden, auch unser Arbeitsplatz ist oftmals nicht mehr nur an einem Ort. Vor allem dem technischen Fortschritt haben wir es zu verdanken, dass wir uns auf dem gesamten Erdball bewegen können und die ganze Welt binnen Stunden erreichbar ist. Heute arbeiten wir in Wien, morgen halten wir eine Präsentation in Toronto. Dazwischen liegen nicht nur Tausende Kilometer, sondern auch viel Arbeit, welche oft unterwegs verrichtet werden muss.

Und Teamwork ist dabei meist ein sehr wichtiger Faktor. Große Projekte verlangen das Know-How und die Arbeitskraft vieler Individuen, Manager müssen mit ihren Mitarbeitern stets in Kontakt bleiben um über den aktuellen Stand informiert zu sein und neue Aufgaben verteilen zu können.

Es gibt verschiedene Möglichkeiten diese Arbeiten aus der Ferne durchzuführen. Telefon- bzw. Videokonferenzen sind heutzutage in vielen Unternehmen an der Tagesordnung. Eine reine Sprachverbindung reicht allerdings meist nicht aus, um etwas zu besprechen, zu erklären oder zu präsentie-

ren. Obwohl eine Videoverbindung bereits mehr Interaktion zwischen den Gesprächsteilnehmern ermöglicht und Emotionen und Empfindungen besser vermitteln kann, sind gemeinsame Arbeiten nur sehr schwer möglich. Liegen Dokumente, Dateien oder nur handschriftliche Notizen vor, so ist es meist problematisch, diese einfach auszutauschen.

1.2 Ausgangsszenarien und Zielsetzung

Nicht nur der Austausch der Informationen beider Gesprächspartner stellt ein Problem dar, sondern auch die gemeinsame Arbeit mit einem Computer stellt eine Hürde dar. Genau an diesem Punkt setzt diese Masterarbeit an. Vor allem für kollaborative Arbeiten sind die bestehenden Lösungen ungeeignet und lange Reisen etc. sind notwendig um sich an einem Ort zu treffen um gemeinsam etwas zu erarbeiten.

Ziel dieser Arbeit ist es einen Prototypen zu schaffen, der ein kollaboratives Arbeiten über das Internet ermöglicht. Sprachverbindungen alleine reichen nämlich nur selten aus, wenn die Benutzer Dokumente, Informationen, und Dateien austauschen wollen um diese gemeinsam zu bearbeiten. Durch neuartige Eingabegeräte und Projektionstechnik soll diese Hürde überwunden werden, in dem ein digitales Whiteboard die Schnittstelle zwischen den entfernten Benutzern erweitert. Zusätzlich sollte eine Online-Konferenz durch diverse Funktionalitäten dem Benutzer attraktiver gemacht werden, indem z. B. sofort eine digitale Mitschrift zur Verfügung gestellt wird oder Daten zentral auf einem Server gespeichert und gesichert werden.

1.3 Gliederung der Arbeit

Nach diesem Einleitungskapitel beschäftigt sich diese Masterarbeit mit Visionen kollaborativen Arbeitens in nächster und auch ferner Zukunft. Danach wird der aktuelle Stand der Technik betrachtet. Hierbei wird auf *Multi-Touch* basierende Oberflächen, sowie auf Anwendungen welche bereits heute gemeinsames Arbeiten über das Internet ermöglichen, eingegangen.

Im Anschluss daran wird ein Konzept für einen neuen Prototypen erstellt, bei dem eine mit digitalen Stiften bedienbare Oberfläche mit einer kollaborativen Anwendung verknüpft werden soll. Basierend auf diesem Konzept wird in Kapitel 5 auf die zu verwendenden Technologien eingegangen.

Auf diesem Konzept aufbauend wird die Implementierung des Prototypen erläutert. Hierbei soll gezeigt werden, inwiefern die Implementierung eine gewisse Flexibilität zulässt, Konzepte und Ideen zu testen. Im Anschluss daran folgt die Evaluierung der prototypischen Anwendung, bei der verschiedene Benutzer die Anwendung testen. Die abschließende Zusammenfassung soll auch einen Ausblick in die Zukunft dieser Thematik geben.

Kapitel 2

Visionen für das gemeinsame Arbeiten in der Zukunft

Als im Jahre 2002 *Minority Report* [17] in die Kinos kam, konnten sich nur wenige vorstellen, dass derartige futuristische Benutzeroberflächen (siehe Abb. 2.1), welche mit Händen und Gesten bedienbar sind, einmal Wirklichkeit werden. Berührungssensitive Oberflächen, durch die man auch noch hindurchsehen kann, erinnern auch heute noch eher an einen *Science-Fiction* Film.

In Museen oder bei künstlerischen Ausstellungen (z. B. *Ars Electronica*¹, Linz) wurden schon seit längerem immer wieder Schnittstellen vorgestellt, bei denen z. B. mit Sprache, Pusten oder Klopfen interagiert werden kann. Doch dabei handelt es sich meist nicht um eine sinnvolle Verwertung im Sinne einer Arbeitsanwendung, wie sie tagtäglich in Büros oder Konferenzräumen eingesetzt wird.

¹www.aec.at

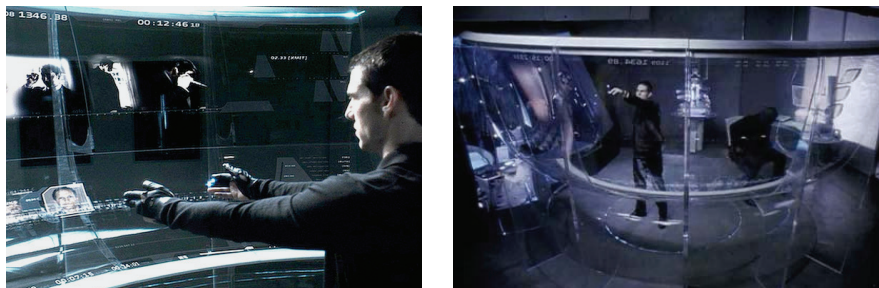


Abbildung 2.1: Der berührungssensitive Bildschirm aus *Minority Report* ermöglicht dem Protagonisten seine Arbeit zu verrichten, aus [17].

2.1 Digitale Wände im Einsatz

In diversen Zukunftsvisionen, vor allem in Demonstrations-Videos, wird gezeigt wie digitale Oberflächen, meist Tafeln und Wände, unser Zuhause und Arbeitsumfeld verändern können.

Das digitale Wohnzimmer

Man stelle sich vor, anstelle einer Tapete eine digitale Wand einzusetzen und diese von einem Computer aus anzusteuern. Man könnte nicht nur jeden Tag andere Fotos bzw. Posters per „Klicken und Ziehen“ an die Wand hängen, die Farbe ändern, sondern auch noch Videos und Fernsehprogramme direkt in einem beliebigen Bereich der Wand ansehen. Auch wenn man ein Buch liest, kann diese Wand z. B. die Stimmungen des Buches wiedergeben, sei es durch Farben oder Bilder. Gibt es dann auch noch Möglichkeiten mit dieser Wand zu interagieren, z. B. durch Videokameras oder berührungsempfindlichen Sensoren, dann wäre das digitale Zimmer beinahe perfekt.

Es fehlt nur noch an der Software, die diese Vision erst möglich macht. Dabei sollte keine Maus oder Tastatur benötigt werden um mit seinem „Zimmer“ zu interagieren. Das intelligente Programm könnte selbständig erkennen wer sich im Raum befindet und mit Hilfe dieser Information den Raum gestalten. Die Fotos könnten vom zentralen Familien-Server ausgewählt und im Zimmer präsentiert werden. Die passende Musik wird aufgrund der letzten Bestellungen im Online-Musikladen gesucht und der aktuelle Terminkalender aus einer Kalender-Applikation auf die Wand geladen.

Um sich im Büro wohl zu fühlen können alle beliebigen privaten Fotos und Poster an der Wand hängen – und ein Knopfdruck genügt und die Wand wechselt ihr Aussehen vor einem wichtigen Geschäftstreffen. Die aktuelle Agenda wird nun projiziert und Notizen können direkt auf der Wand gemacht werden. Aufgaben und Termine, welche am digitalen Wandkalender eingetragen werden, können so sofort mit denen der Kollegen synchronisiert werden.

Des Weiteren kann diese Wand auch für Fernkonferenzen oder Besprechungen mit Kollegen genutzt werden, in dem diese als digitales Fenster zum Anderen fungiert.

Krankenhaus der Zukunft

In den Abbildungen 2.2, welche aus einem Video der *Microsoft Industry Innovations Group* entnommen wurde, werden Zukunftsvisionen rund um das Thema Gesundheit und Krankenhaus präsentiert [23]. Einen nicht unbeträchtlichen Teil davon wird auf ein kollaboratives System eingegangen, mit Hilfe dessen der Arzt, der Therapeut und der Patient, jeweils von verschiedenen Orten aus, die Behandlungen und deren Ergebnisse diskutieren.

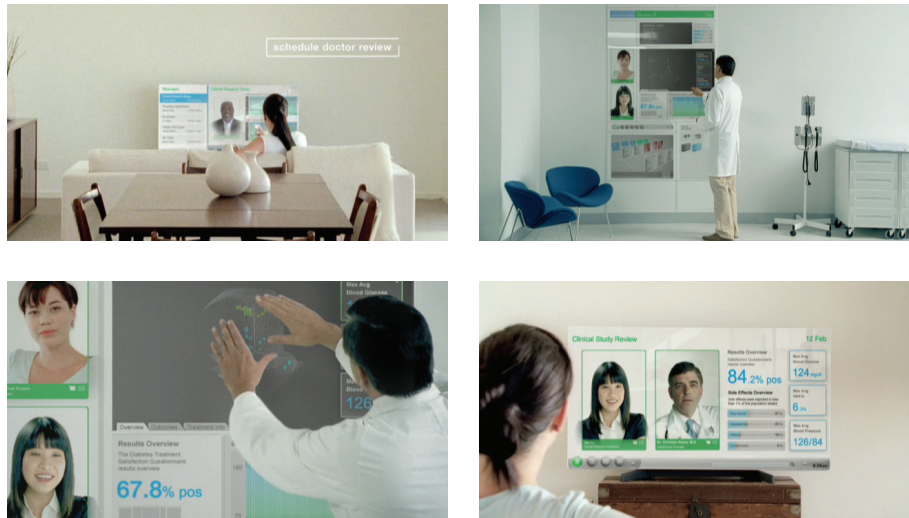


Abbildung 2.2: Der Arztbesuch der Zukunft. Aus [15].

Im Video wird z. B. auch die Vielseitigkeit der digitalen Oberflächen verdeutlicht. Die Patientin nimmt über ihren hoch aufgelösten Fernseher an der Konferenz teil, die Therapeutin über ihr Notebook, und der Arzt steht vor einer digitalen Wand, welche wenige Augenblicke davor noch beruhigende Animationen gezeigt hat und sich erst dann in eine multifunktionelle Arbeits- und Kommunikationsfläche verwandelt hat.

Kommunikation in Industrieprozessen

Eine weitere Demonstration derselben Arbeitsgruppe zeigt Anwendungen im Umfeld eines großen Automobilkonzernes. Dabei wird die Zusammenarbeit in einem Innovationsprozess dargestellt: Eine Design-Gruppe in Nord-Amerika muss dabei mit dem Zulieferer aus Asien zusammenarbeiten. Diese Ergebnisse werden dann von einem *OEM*² produziert bzw. zusammengesetzt. Alle Prozesse laufen dabei computerunterstützt und computergesteuert ab.

Wie man in Abbildung 2.3 erkennen kann, befindet sich in der Design-Abteilung (links oben) eine digitale Wand, welche die Ergebnisse, die die einzelnen Mitarbeiter bearbeiten, sofort präsentiert (rechts oben). Darüber hinaus bietet diese Wand auch die Möglichkeit, wie durch ein Fenster mit anderen Mitarbeitern oder Kollegen direkt zu interagieren. So wird der in den beiden unteren Abbildungen dargestellte dreidimensionale Autositz von allen Seiten betrachtet. Darüber hinaus können sich die beiden Geschäftspartner über eine hochauflösende Video- und Tonverbindung unterhalten.

²Original Equipment Manufacturer, englisch für Originalausrüstungshersteller.

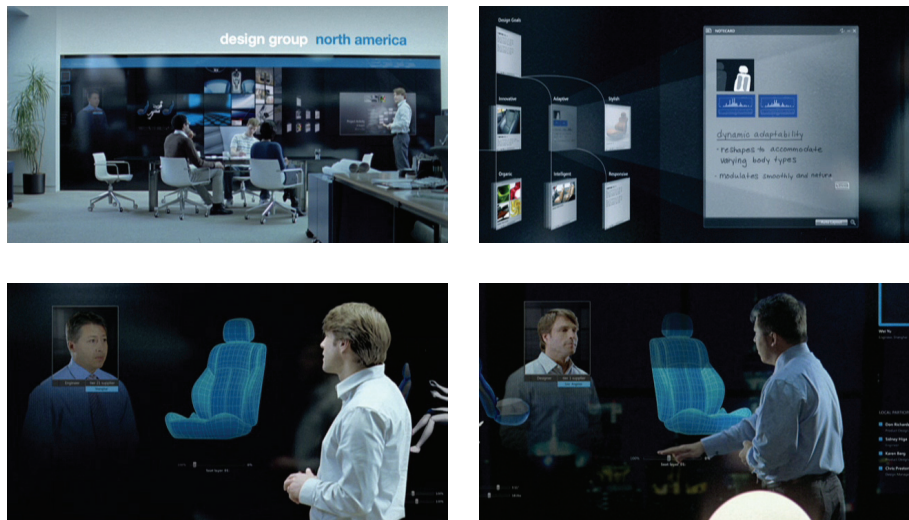


Abbildung 2.3: Vision einer über Kontinente verstreuten Design Abteilung einer Automobilfirma. Aus [16].

Home Entertainment

In [9] machen sich zwei Autoren über die Zukunft von *Apple* Gedanken und „erforschen“ deren zukünftige Produktpaletten. Hierbei zeigt sich sehr deutlich, dass es vor allem um Produkte gehen wird, welche das Heim immer mehr vernetzen. Unter den Produkten, welche 2013 auf den Markt kommen sollen, befinden sich ein vernetzter Multimedia-Radiowecker und ein berührungssensitiver Bilderrahmen, welcher ebenfalls über eine Internetverbindung verfügt. Ebenfalls wird vermutet, dass *Apple*, welche bereits mit dem *iPhone* ein neuartiges *Multi-Touch*-Produkt an die Endkunden brachte, an einer Antwort auf *Microsofts Surface* (siehe Abschnitt 3.2.1) arbeitet.

2.2 Vernetzte Welten

Heutzutage bilden Konzerne und Firmen ein weltumspannendes Netz an Zweigstellen, Niederlassungen und lokalen Dienststellen. Durch die Globalisierung ist dies nicht nur ein notwendiger Schritt sondern ein vor allem auch geschäftsbringender Vorteil großer Firmen. Vor allem Geschäftsreisende verbringen viel Zeit auf Flughäfen und in Hotels, von denen sie ebenso aus arbeiten müssen, wie aus dem eigenen Büro.

In einem weiteren Video [12] stellte *Microsoft* 2005 eine Konzept für eine „Real-Time Collaboration“ Software dar. Dabei sitzt ein Geschäftsreisender am Flughafen in Chicago fest, da sein Flug gestrichen wurde. Er sollte bereits nur wenige Stunden später ein Seminar vor 700 extra nach Washington D.C.

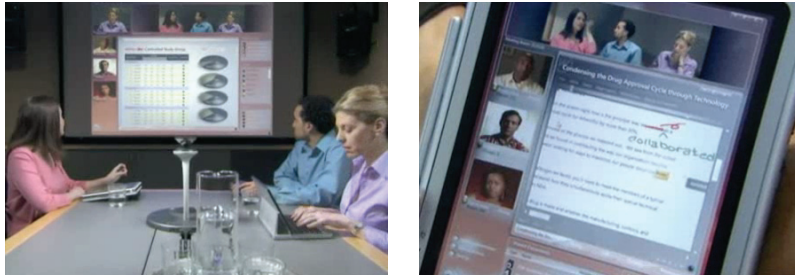


Abbildung 2.4: „Real-Time Collaboration Vision“ für 2010. Aus [12].

angereisten Wissenschaftlern halten, welches er aber so nicht mehr schaffen konnte. Im Video wird eine Vision einer kollaborativen Videokonferenzlösung gezeigt (siehe Abb. 2.4), mit welcher es zuerst möglich war mit einer anderen Kollegin den Vortrag gemeinsam auszuarbeiten und ihr sodann einen Teil des Vortrags zu übergeben. Im letzten Schritt sprach er aber vom Hotelzimmer aus zu den 900 Zuhörern.

Telearbeit

Nicht nur geringere Bürokosten für den Arbeitgeber, sondern auch weniger Treibstoffverbrauch für Arbeitnehmer und Umwelt machen das Arbeitsmodell der Telearbeit immer interessanter. Dennoch ist es vor allem bei Teamarbeiten nur sehr schwer möglich sinnvoll mit Telearbeitern zusammenzuarbeiten. Neuartige und moderne Applikationen, die einen solchen Arbeitsprozess ermöglichen, unterstützen und auch kontrollieren können, wären dazu eine Lösung.

Aber nicht nur für klassische Fernarbeiter wäre eine solche Applikation interessant. Wie schon in obiger Vision beschrieben hält sich unsere Gesellschaft immer öfters auf Flughäfen, in Hotel und im Transit von und zur Arbeit auf. Vielleicht könnte mit Hilfe neuartiger kollaborativer Kommunikationstechnik die Produktivität gesteigert werden.

Technische Realisierbarkeit

Technisch gesehen sind die meisten Grundlagen für diese Visionen bereits heute geschaffen, und einzelne Teilanwendungen sind bereits umgesetzt. Immer höhere Datenkapazitäten und schnellere Datenleitungen ermöglichen einen unbegrenzten Datenaustausch auf der ganzen Welt. Man ist immer weniger an Festnetzleitungen gebunden, denn vor allem der mobile Ausbau geht in großen Schritten voran.

Kapitel 3

Stand der Technik und verwandte Arbeiten

In diesem Kapitel werden aktuelle und auch frühere, ähnliche Arbeiten betrachtet. Zuerst wird auf bereits vorhandene vernetzte Systeme eingegangen, die es ermöglichen miteinander zu kollaborieren. Danach werden aktuelle, digitale Oberflächen, seien es Whiteboards, Tische oder ganze Wände gezeigt.

3.1 Verteiltes kollaboratives Arbeiten

Schon früh haben sich Wissenschaftler mit kollaborativen Systemen beschäftigt und dabei standen vor allem Videokonferenzsysteme im Vordergrund. Es wurden aber auch Applikationen zum gemeinsamen Schreiben oder Zeichnen entwickelt.

3.1.1 Erste Videokonferenzsysteme

Schon 1991 haben Tang et al. in *VideoWhiteboard* [26] versucht auf zwei verschiedenen Bildschirmen gemeinsam zu arbeiten. Dabei wurde das Bild des einen Benutzers auf den Schirm des Anderen per Video übertragen. Ein auftretendes Problem ist, nicht zu wissen, wo bzw. an was der Andere gerade arbeitet. Dies wurde in dieser Arbeit zu lösen versucht. Dabei wurden die Schatten der Handposen des Gegenübers mit einer Videokamera aufgenommen und ebenfalls am Bildschirm dargestellt.

Auch *MAJIC (Multiparty Video Conferencing at Virtual Social Distance)* [19] beschreibt ein Videokonferenzsystem, bei dem es erstmals möglich ist, bei drei Teilnehmern, welche sich alle an einem anderen Ort befinden, sich jemandem zuzuwenden und ihm in die Augen zu sehen. Dazu werden Bilder auf einer gekrümmten, halbdurchlässigen Projektionsfläche projiziert. Die Videobilder werden jeweils von zwei Kameras aufgenommen, nämlich für jeden der beiden weiteren Teilnehmer separat (siehe Abb. 3.1). Somit ist es

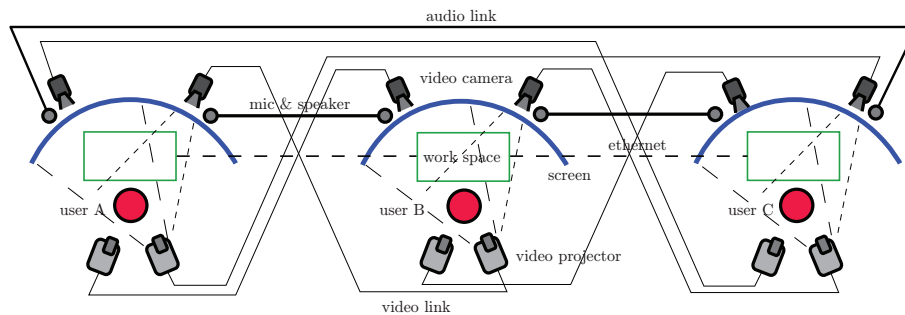


Abbildung 3.1: Technischer Aufbau des Videokonferenzsystems MAJIC. Aus [19] (bearbeitet).

für Teilnehmer A möglich, dem Teilnehmer B in die Augen zu sehen, wenn er in die rechte Kamera sieht und dabei zeichnet aber die linke Kamera die Seite seines Gesichtes auf. Für Teilnehmer B entsteht der Eindruck, dass er angesehen wird, für Teilnehmer C blickt Teilnehmer A in die Richtung von Teilnehmer B.

Leider ist diese gute Technik nur schwer skalierbar und auf drei bis vier einzelne Konferenzteilnehmer beschränkt. Des Weiteren ist ein sehr aufwändiger Hardware-Aufbau nötig.

3.1.2 Chats, Instant Messaging und VoIP

Internet Chats: Emoticons und Avatare

Chats sind seit Beginn des Internets ein wesentlicher Bestandteil der Online-Welt. Viele Internet-Portale bieten Räume für Gleichgesinnte an, in denen geredet und diskutiert wird und man sich ausgetauscht kann. Um in der schriftlichen, elektronischen Kommunikation Stimmungs- und Gefühlszustände auszudrücken, entwickelten sich in Chats und Foren *Emoticons*, welche aus normalen Satzzeichen bestehen und z. B. einen *Smiley* – :-) – nachbilden.

Eine Weiterentwicklung von Emoticons sind Avatare. Diese repräsentieren Personen durch fiktive Menschen, Tier- oder Fantasiegestalten in virtuellen Welten. Die Benutzer können das Aussehen und Verhalten dieser beeinflussen und sich so in einer Online-Gemeinschaft darstellen. Vor allem in Online-Spielen wie *SecondLife*¹ (siehe Abb. 3.2) oder *World of Warcraft*² spielen Avatare eine große Rolle und können dabei nicht nur die tatsächliche Gestalt eines Benutzers darstellen, sondern auch seine Persönlichkeit und Gedanken repräsentieren.

¹www.secondlife.com, 13 Millionen Einwohner (April 2008)

²www.worldofwarcraft.com, 10 Millionen Spieler (Jan. 2008)



Abbildung 3.2: Avatare in SecondLife.

Instant Messaging

Heutzutage nehmen immer mehr *Instant Messaging* Programme, seien es *AIM*³, *Jabber*⁴, *Yahoo Messenger*⁵, *ICQ*⁶ oder *Skype*⁷ unsere Computer ein. Diese ermöglichen es sofort zu sehen, welcher Freund oder Geschäftspartner ebenso vor seinem Computer sitzt und Zeit zum Kommunizieren hat. Was diese Applikationen so beliebt gemacht hat (vgl. Benutzerzahlen in der Fußnote), ist vor allem deren einfache Bedienung. Es sind meist keinerlei Einstellungen bezüglich der Netzwerkverbindung notwendig.

Sprach- und Videotelefonie

Microsoft Netmeeting, welches bereits in *Windows 95* integriert wurde, ist eines der ersten frei verfügbaren *VoIP*⁸ und Videokonferenzsysteme. Firewalls sind die großen Hürden derartiger Programme, da (möglichst) direkte Verbindungen zwischen den Rechnern hergestellt werden müssen. So war es oftmals notwendig Firmennetzwerke und Router derart zu konfigurieren, damit die richtigen Ports weitergeleitet werden. *Skype* schaffte den Durchbruch, indem die Entwickler eine Technologie hervorbrachten, welche es ermöglicht sogar durch abgesicherte Firmennetzwerke gut funktionierende *VoIP*-Verbindungen aufzubauen [2].

Neben Sprachverbindungen und Videostreams bot bereits *Netmeeting* ei-

³www.aim.com, 53 Millionen aktive Benutzer (Sept. 2006)

⁴www.jabber.com, 40 Million aktive Benutzer (Jan. 2007)

⁵messenger.yahoo.com, 22 Millionen Benutzer (Sept. 2006)

⁶www.icq.com, engl. „I seek you“, 15 Millionen Benutzer (Juli 2006)

⁷www.skype.com, 12 Millionen Benutzer online, 276 Millionen registriert (Feb. 2008)

⁸ *VoIP*: *Voice-over-IP*, Sprache wird dabei digital in Datenpaketen über das *Internet Protocol* (IP) übertragen.

ne einfache Zeichenapplikation (vgl. *Windows Paint*) an, mit Hilfe deren einfache Skizzen gezeichnet werden können. *Skype* wiederum bietet eine API an, welche es ermöglicht Zusatz-Anwendungen (vgl. *Skype Extras*⁹) kostenfrei zu entwickeln. Darunter wurden ebenfalls mehrere einfache kollaborative Zeichenapplikationen entwickelt, z. B. *Whiteboard Basic* (siehe Abb. 3.3 (a)).

3.1.3 Online-Textverarbeitung und *Desktop Sharing*

Wikis

Ein *Wiki* ist eine Sammlung von Webseiten, welche allerdings von jedem erstellt, editiert oder gelöscht werden können. Das auf diese Art und Weise entstandene Werk *Wikipedia*¹⁰ ist die größte und umfangreichste Enzyklopädie der Welt mit über 2,5 Millionen Artikel (Stand: Juni 2008). An der deutschsprachigen Ausgabe arbeiten über 7.000 Autoren mit, welche ihre Texte unter der GNU-Lizenz¹¹ bereitstellen und somit *Wikipedia* zu einer sogenannten „freien Enzyklopädie“ machen.

Die Software dahinter ist eine besondere Art eines *Content-Management-Systems*¹², welches auf der einen Seite den vollen Zugriff aller erlaubt, auf der anderen Seite aber auch dafür sorgt, dass die Gemeinschaft die volle Kontrolle über das Werk hat. Somit werden Entscheidungen über Artikel gemeinsam gefällt oder ungewünschte Änderungen wieder rückgängig gemacht.

Office Anwendungen im Web

Im August 2005 startet die Softwarefirma *Upstartle* ein web-basiertes Textverarbeitungsprogramm namens *Writely*, welches im September 2006 von *Google* übernommen und in *Google Text & Tabellen*¹³ integriert wurde. Mit dieser Online-Plattform ist es möglich gemeinsam an Textdokumenten, Tabellenkalkulationen und auch Präsentationen zu arbeiten. Dazu kann man sich mit einem *Google*-Konto anmelden und Dateien für andere *Google*-Benutzer sichtbar bzw. auch bearbeitbar machen.

Des Weiteren ist es sogar möglich, dass mehrere Autoren zeitgleich am selben Dokument arbeiten. Dazu wird in einem bestimmten Intervall bzw. nach jedem Absatz das Dokument neu ausgetauscht und die anderen Bearbeiter können die Änderungen sofort sehen.

Kritisiert wird, dass *Google* Zugriff auf alle gespeicherten Daten hat und dies ein Sicherheitsrisiko für vertrauliche Daten ist. Dies macht dieses Produkt für Firmenkunden uninteressant.

⁹extras.skype.com, developer.skype.com

¹⁰www.wikipedia.com

¹¹GNU-Lizenz für freie Dokumentation, <http://www.gnu.org/licenses/fdl.html>

¹²Ein *Content-Management-System* erlaubt es einem Benutzer Inhalte einer Web Seite im laufenden Betrieb zu ändern. HTML oder Server Kenntnisse sind dabei nicht nötig.

¹³docs.google.com

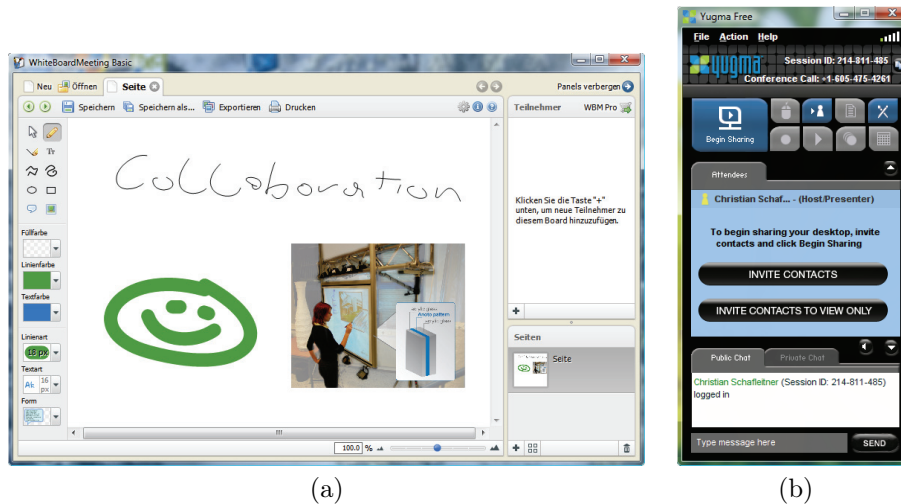


Abbildung 3.3: Kollaborative Desktop-Anwendungen: (a) *Whiteboard Basic*, welches auf Skype basiert, und die *Yugma*-Kontrolloberfläche (b).

Desktop Sharing und Präsentationen

*Yugma*¹⁴ ist eine kommerzielle Software, die es erlaubt seinen eigenen Desktop und seine Anwendungen mit anderen Benutzern über das Internet zu teilen, und somit Präsentationen abzuhalten oder Anwendungen zu demonstrieren. Dabei kann die Kontrolle des eigenen Computers auch an andere Benutzer abgegeben werden. Ein Problem dieser Systeme ist, dass meist lediglich eine Person die Steuerung übernehmen kann, und die anderen nur passiv zusehen können. Dennoch bieten derartige Anwendungen viel Potenzial um zu kollaborativen Systemen ausgebaut zu werden. In Abbildung 3.3 (b) wird die Kontrolloberfläche von *Yugma* gezeigt.

Auch *SmartTechnologies*, der Marktführer digitaler Whiteboards (siehe auch Abschnitt 3.2.3), bietet eine sehr ähnliche Lösung wie *Yugma* an. *BridgeIt*¹⁵ erlaubt es ebenfalls auf einfache Art und Weise den Desktop für Andere freizugeben um so gemeinsam an Dokumenten zu arbeiten. Neben einer eingebetteten *VoIP*- und Videokonferenzlösung gibt es auch eine Whiteboard-Anwendung.

3.2 Digitale Arbeitsflächen

Interaktive und digitale Oberflächen nehmen immer mehr unseren Arbeits- und Büroraum ein. Seien es Monitore, Projektionsflächen, berührungssensitive Oberflächen oder neuartige Displays, welche diese Technologien kombi-

¹⁴www.yugma.com

¹⁵<http://www2.smarttech.com/st/en-US/Products/Bridgit/>

nieren und bereits z. B. in Mobiltelefone verbaut wurden. Vor allem diese bieten eine neue Möglichkeit einfach und intuitiv mit einer Computeranwendung zu interagieren und so Aufgaben leichter, oder auch nur interessanter, zu lösen.

Bei diesen Projekten geht es sehr oft darum neue Eingabemöglichkeiten zu entwickeln bzw. zu testen. Die letzten 20 Jahren dominierten Maus und Tastatur beinahe alle Applikationen. Für interaktive Anwendungen sind diese Eingabegeräte, welche sich am Desktop-Rechner durchgesetzt haben, aber in der Regel keine gute Lösung. Obwohl die meisten Menschen mit diesen sehr gut vertraut sind, schaffen Maus und Tastatur oftmals eine zusätzliche Hürde, da jeder auf sein eigenes Eingabegerät fixiert und beschränkt ist.

Ein weiteres Problem bei neuen oder mehreren Eingabegeräte ist, dass Betriebssysteme in der Regel nur *eine* Tastatur und Maus unterstützen (dazu siehe auch Abschnitt 5.2.3). Wenn es also darum geht, dass mehrere Personen gleichzeitig mit einer Installation oder einer Anwendung interagieren können meist keine Standard-Programm-Frameworks verwendet werden.

Kollaboratives Arbeiten

Während ein Benutzer vor seinem Computer meist nur alleine arbeitet, oder sich lediglich über das Internet mit anderen Benutzern austauscht, ermöglichen solche digitale und interaktive Oberflächen ein simultanes und gemeinsames Arbeiten auf nur einem Arbeitsplatz. Durch die Anordnung bzw. den Aufbau der technischen Gerätschaften als Tisch oder Wand entsteht bereits automatisch eine kommunikativere Arbeitsumgebung. Im Gegensatz dazu wirkt ein Computer, vor dem man zu zweit oder zu dritt sitzt, eher als zusätzliche Hürde.

In den folgenden Projekten und Produkten werden Finger, Stifte und intelligente Objekte (*SmartObjects*) als Eingabemedien benutzt.

3.2.1 Interaktiver Tisch: *Microsoft Surface*

*Surface*¹⁶ ist ein interaktiver Tisch, in dem ein auf *Windows Vista* basierender Computer, sowie eine berührungssensitive Oberfläche verbaut wurde (Abb. 3.4 (a)) [14]. Bereits 2004 wurde von Andy Wilson (*Microsoft Research*) ein Prototyp namens *TouchLight* [29] entwickelt, auf welchem das in *Surface* eingesetzte Fingertrackingverfahren basiert. Dieses verwendet mehrere Kameras um sowohl Finger als auch Objekte zu erkennen.

Jun Rekimoto war einer der ersten, der z. B. in [22] einen Multi-Touch Ansatz erläuterte. Nicht nur das *iPhone*¹⁷ von *Apple*, sondern auch *Surface* basieren auf Rekimotos ersten Benutzbarkeitsstudien und deren Ergebnissen hinsichtlich der Interaktion mit mehreren Fingern bzw. Händen.

¹⁶www.microsoft.com/surface

¹⁷www.apple.com/iphone

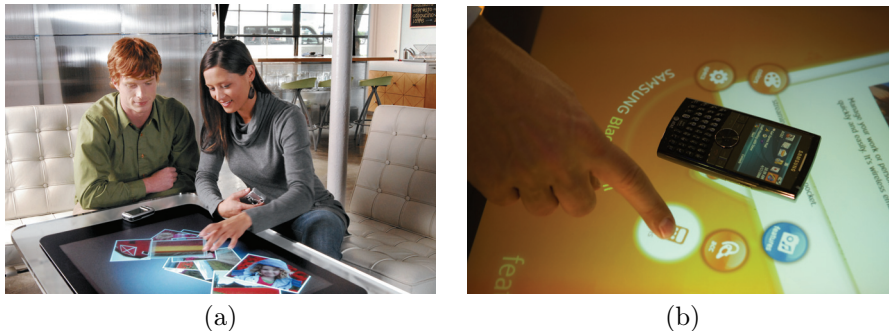


Abbildung 3.4: *Microsoft Surface*: (a) Multi-Touch Demo-Applikation mit Bildern, (b) *AT&T* Promotion und Verkaufsanwendung für Mobiltelefone. Aus [14].

Der *Surface*-Tisch wurde in erster Linie für Unterhaltungsangebote in der Tourismus- und Telekommunikationsbranche entwickelt. Die ersten Demoapplikationen erlauben dem Benutzer Speisen zu bestellen und zu bezahlen, eigene Sportgeräte zu gestalten oder Fotos, welche auf einer Digitalkamera oder einem Mobiltelefon gespeichert sind, einfach untereinander auszutauschen. Dabei kann man eine weitere Funktionalität von *Microsoft Surface* erkennen: einfacher und berührungsloser Datenaustausch über kabellose Netzwerke zwischen unterschiedlichen Geräten. Somit lässt sich der interaktive Tisch durch Ablegen unterschiedlicher Gegenstände, welche durch das optische Trackingverfahren erkannt werden, steuern.

Für Privatpersonen oder kleine Unternehmen ist der Tisch (noch) zu teuer, er wird für ca. 10.000 US-Dollar lediglich an Partnerunternehmen von Microsoft verkauft. Der erste kommerzielle *Surface*-Tisch wurde im Mai 2008 für *AT&T* mit einer Verkaufs- und Promotionsoftware (siehe Abb 3.4 (b)) für Mobiltelefone ausgeliefert und kann nun in diversen *AT&T* Geschäften genutzt werden.

3.2.2 Digitale Schultafel: *eChalk*

eChalk [7] ist eine elektronische Kreidetafel, welche an der Freien Universität Berlin entwickelt wurde. Ziel war es, die handschriftlichen Notizen des Vortragenden digital zu erfassen und mit Bildern oder interaktiven Applikationen (z. B. physikalischen Simulationen) zu kombinieren. Dabei kann eine Vorlesung auch über das Internet mitverfolgt werden, bzw. im Nachhinein abgespielt werden. Des Weiteren generiert *eChalk* auch automatisiert eine *Adobe PDF*-Datei mit den Inhalten der Tafel, dadurch ist ein Mitschreiben der Studenten nicht mehr nötig.

Diese elektronische Tafel ist vor allem für den Physik und Mathematik-Unterricht ausgelegt. So wurde z. B. weitere Entwicklungsarbeit betrieben,

um mathematische Formeln und Notationen, welche handschriftlich erfasst wurden zu erkennen [27].

eChalk soll die Benutzung einer Tafel für Professoren und Lehrende wieder attraktiver machen. Durch Folien bzw. *PowerPoint* Präsentationen tendieren Vortragende dazu viel zu schnell vorzugehen und die Studenten sind dadurch oft überfordert. Ein Vortrag auf einer Tafel hingegen benötigt länger und somit haben die Studenten genug Zeit um mitzudenken und den Stoff zu verarbeiten.

Das komplette System ist allerdings ausschließlich auf die Benutzung als Vortragssystem ausgelegt, ein kollaboratives Arbeiten ist darum nicht möglich. Mit der digitalen Wand kann lediglich eine Person gleichzeitig interagieren.

3.2.3 Digitales Whiteboard: *Intoi*

*Intoi*¹⁸ ist ein digitales Whiteboards, auf dem es möglich ist, dass mehrere Benutzer zeitgleich und simultan mit digitalen Stiften auf einer Oberfläche arbeiten [4]. Das Besondere an *Intoi* ist, dass die gesamte Benutzerinteraktion auf die Bedienung mit einem bzw. mehreren Stiften angepasst wurde. Die Bedienung eines klassischen *WIMP*¹⁹-Interfaces, wie es alle gängigen Betriebssysteme umsetzen, ist mit einem stiftbasierenden Medium nicht nur ungewohnt, sondern meist auch umständlich. Bei *Intoi* wurde deshalb ein neues Konzept gewählt.

Es sollte jedem, der ein analoges Whiteboard bedienen kann, ermöglicht werden mit *Intoi* zu arbeiten. Dazu war es notwendig jegliche Zusatzfunktionen neben dem gewohnten Zeichnen leicht erlern- und auswählbar in das System zu integrieren. Dies geschieht z. B. mit Hilfe einer „Fernbedienung“, auf welcher die wichtigsten Werkzeuge (Stift, Radierer, Auswahl, *Zoom/Pan*) auswählbar sind. Auch der Wechsel zwischen Farben ist durch die „Fernbedienung“ einfach möglich. Durch die drucksensitive Technik des Stiftes ist es für den Benutzer intuitiv möglich einen Strich in verschiedenen Stärken zu zeichnen.

Alle in *Intoi* erstellten Skizzen oder handschriftlichen Notizen werden als Vektorgrafiken gespeichert. Eine Eingabemöglichkeit per Tastatur existiert nicht. Durch die Fehleranfälligkeit von Handschrifterkennungssystemen wurde auf diese verzichtet. Des Weiteren können zuvor erstellte Grafiken und Bilder, als auch *PowerPoint*-Präsentationen oder *PDF*-Dokumente in ein *Intoi* Projekt importiert werden.

¹⁸engl. *Interchange of Ideas* - Gedankenaustausch

¹⁹Abk. für ein grafische Benutzerumgebung (*GUI*) mit *Windows*(Fenster), *Icons*, Menüs und *Maus-Pointer*(Zeiger).



Abbildung 3.5: *Intoi*: Nicht nur auf einer grenzenlosen Zeichenfläche kann geschrieben werden, auch auf Bilder können Anmerkungen gemacht werden.

Grenzenlose Zeichenfläche

Das Kernstück der Applikation ist eine grenzenlose Zeichenfläche, welche dem Benutzer soviel Platz bietet, wie er auch wirklich benötigt. Somit kann es nicht mehr passieren, dass auf einmal der Platz ausgeht oder man über den Rand hinaus schreiben muss. Die Zeichenfläche kann beliebig verkleinert oder vergrößert und verschoben werden. Ein Hilfsraster dient zur Orientierung und unterstützt dem Benutzer beim Schreiben und Skizzieren. Da *Intoi* mit Vektorgrafiken arbeitet entsteht beim Vergrößern der Zeichenfläche kein Qualitätsverlust der auf *Intoi* angefertigten Skizzen und Zeichnungen.

Curving vs. Scrolling

Intoi setzt auch ein neuartiges Menüsystem um: Auf Basis eines *Piemenus*²⁰ [5] wurden drei kontextbezogene Menüs entwickelt. In Abbildung 3.6 (a) ist das Standardmenü mit der geöffneten Farbauswahl dargestellt. Die sensitive Fläche in der Mitte des „Donut“-förmigen Menüs kann des Weiteren auch zur Gestenerkennung verwendet werden. Wie man in Abbildung 3.6 (b) erkennen kann, können in diesem Bereich auch Dialoge angezeigt werden.

Als besonders störend wurde bei diversen Benutzertests das Bedienen einer Scrollbar mit einem Stift empfunden. Daher wurde trotz grenzloser Zeichenfläche und Dateilisten auf Scrollbars verzichtet. Anstatt dessen wurde ein *CurveDial* [24] implementiert. Mit Hilfe dessen ist es möglich durch ein-

²⁰Unter *Piemenus* versteht man kreisförmige und kontextbezogene Menüs, wie sie zum Beispiel in Computerspielen, aber auch in 3D Animationsprogrammen (z. B. *Alias Maya*) eingesetzt werden.

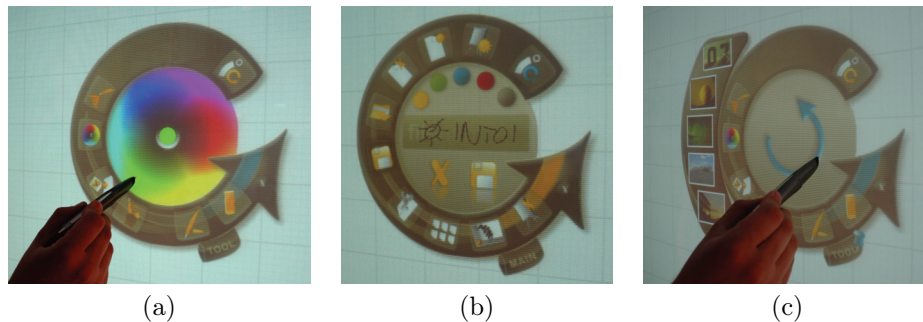


Abbildung 3.6: *Intoi* Menü: Farbauswahl (a) und Dialogfeld (b) innerhalb des *Pie*-Menüs; durch Kreisbewegungen (c) können die Inhalte der Auswahlbox (links) durchgeblättert werden.

fache Kreisbewegungen, abhängig von Radius und Geschwindigkeit, Parameterwerte (z. B. den Zoomlevel oder die Position in einer Liste) zu verändern (siehe auch Abb. 3.6 (c)).

Mehrbenutzerfähigkeit

Des Weiteren wurde die Applikation darauf ausgelegt, dass mehrere Benutzer gleichzeitig und miteinander auf der digitalen Oberfläche arbeiten können. Mit Hilfe eindeutiger Kennnummern der Stifte werden für jeden Benutzer eigene Menüs angelegt und deren Einstellungen gespeichert. Diese Menüs können an jeder beliebigen Position, also genau an der aktuellen Stiftposition des Benutzers, aufgerufen werden.

Intoi erlaubt es bis zu acht Benutzern bzw. Stiften gleichzeitig mit dem Flipchart zu interagieren. Dieses Limit wird allerdings nur durch die Beschränkung der *Bluetooth* Hardware gesetzt, da es lediglich möglich ist, acht *Bluetooth* Geräte an einem *Dongle* zu betreiben.

.NET und *OpenGL*

Das gesamte Projekt wurde in C# und dem *.NET Framework* umgesetzt. Dadurch wurde eine sehr hohe Stabilität der Software erreicht, da der gesamte Quellcode als *managed code*²¹ vorliegt.

Mit Hilfe des *TAO Frameworks*²² wurde es dennoch ermöglicht die gesamten Vektor- und Pixelgrafiken hardwarebeschleunigt zu rendern. Das *TAO*

²¹In C# wird das Allokieren und Freigeben von Speicherbereichen durch eine virtuelle Maschine (*CLR - Common Language Runtime*) erledigt. Ein *Garbage Collector* löscht nicht mehr benötigte Objekte und gibt so den Speicher wieder frei.

²²<http://www.taoframework.com/>

Framework bietet unzählige *Wrapper*²³ Klassen um die in C/C++ implementierten Grafikbibliotheken (u. a. *OpenGL*, *DevIL* und *Cg*) in C# bzw. mit dem *.NET Framework* zu verwenden.

²³Ein Wrapper/Adapter ist ein Entwurfsmuster um eine Schnittstelle zu Programmcode oder Klassen zu schaffen, welche sonst inkompatibel wären.

Kapitel 4

Entwurf und Konzept

Kapitel 2 lieferte einen Überblick über mögliche Visionen einer kollaborativen Arbeitsumgebung in nächster Zukunft, das darauf folgende Kapitel 3 zeigte bereits vorhandene Anwendungen. Hierbei kann man erkennen, dass bereits einige Visionen beinahe verwirklicht wurden, aber es dennoch große Einbußen hinsichtlich Funktionalität, Einheitlichkeit und Bedienbarkeit gibt.

In diesem Kapitel wird ein kollaborativer Workflow untersucht und auf diesen basierend ein Konzept für eine vernetzte Whiteboard-Anwendung erstellt.

4.1 Kollaborative Arbeitsprozesse

Grundsätzlich kann man einen kollaborativen Arbeitsprozess in folgende Teile gliedern.

1. Festlegung des Themas und des Termins
2. Zusammenstellung/Einladung der Teilnehmer
3. Zusammentreffen der Teilnehmer
4. Aufgabenstellung/Gemeinsame Zieldefinition
5. Kreativer und/oder wissenschaftlicher Arbeitsprozess unter Beachtung der Zieldefinition
6. Abschluss der Arbeit
7. Zusammenfassen der Ergebnisse und Feedback
8. Auswertung, Versenden der Ergebnisse an alle Teilnehmer

Dabei spielt es keine Rolle, ob diese Arbeit nun gemeinsam während eines Meetings oder verteilt über eine Computer-Anwendung verrichtet wird. Dennoch wird in diesem Abschnitt auch auf deren Unterschiede eingegangen.

Analyse des Workflows

Die ersten beiden genannten Punkte laufen bereits heute meist automatisiert und computergestützt ab. So z. B. über diverse Online-Kalender oder einen firmeneigenen *GroupWare Server* (z. B. *Microsoft Exchange*), mit dessen beispielsweise E-Mails verwaltet und gefiltert, gemeinsame Zeitpläne erstellt und Termine vereinbart werden können. Man kann über das System zu einem Termin zusagen oder diesen verschieben bzw. ablehnen.

Bei Punkt 3 – dem Zusammentreffen der Teilnehmer – gibt es ebenfalls ein meist strukturiertes Vorgehen. Die Teilnehmer werden sich einander vorgestellt und man nimmt seinen Platz ein. Bereits bei diesem Punkt gilt es eine adäquate digitale Umsetzung für eine Applikation zu überdenken. Hier kann gute Software die Hürde nehmen, in dem kurze Profile, Bilder oder auch Avatare der Teilnehmer angezeigt werden.

Während einer Besprechung ist eine Unterstützung im Arbeitsprozess meist nicht gegeben. Papier und Stift als Kommunikations- und Notizmedium werden häufiger eingesetzt als jegliche Computeranwendung. Gerade bei Fernkonferenzen, in denen es meist nur Ton- oder Videoverbindungen gibt, wird dieses Manko besonders stark sichtbar. Es gibt zwar einzelne Systeme, die analoge Inhalte sofort digitalisieren und verbreiten, dennoch hat sich keines dieser Systeme aus Gründen mangelnder Effizienz durchgesetzt. Allerdings gibt es Präsentations-Anwendungen oder z. B. *Desktop Sharing*, welche den eigenen Desktop für Andere freigeben. Aber auch hier fehlt eine einheitliche Verbindung bzw. treten die in Abschnitt 3.1.3 (*Yugma*) erläuterten Probleme auf. Ein weiteres Problem ist es die verschiedenen Teilnehmer, vor allem wenn man sich nicht sehen oder hören kann, zu synchronisieren.

Ein weiterer wichtiger Punkt ist die Protokollierung eines Arbeitsprozesses. Es gibt bereits Anwendungen die Audio- oder Videoprotokolle erstellen können, die allerdings meist wieder von Hand aus ausgewertet werden müssen. Gerade bei aktiven Gruppenarbeiten wäre es wünschenswert auch im Nachhinein feststellen zu können, wie die Ideen zustande gekommen sind.

4.2 Anforderungen an den Prototypen

Erfolgreiche, moderne Software zeichnet sich heutzutage nicht nur durch umfangreiche Funktionalität, sondern auch durch eine gute und durchdachte Bedienbarkeit, aus. Daher ist es notwendig, dass man einen schnellen und einfachen Zugriff auf die Funktionalitäten hat, um z. B. einen kreativen Denkprozess nicht durch ein Hantieren mit der Applikation zu stören.

Der in dieser Arbeit vorgestellte Prototyp sollte ab dem Beginn eines Meetings eingesetzt werden, eine Anbindung an ein *GroupWare* System hinsichtlich Einladungen und Zugriffsrechte wäre eine optionale Erweiterung. Da es schon genug funktionierende Audio- und Videokonferenzanbieter gibt, wird sich der Prototyp auf die Umsetzung einer vernetzten Whiteboard An-

wendung konzentrieren. Schnittstellen zu externen Programmen sollten aber angedacht werden. Mithilfe einer physikalischen Installation soll die Interaktion mit anderen Benutzern, auch über eine größere Distanz hinweg, leichter fallen.

4.2.1 Natürliche Arbeitsweise

Die Anwendung soll möglichst einfach bedienbar sein. Die digitale Wand soll in erster Linie wie eine Schultafel oder ein Whiteboard mit einem Stift bedient werden können.

4.2.2 Anmeldung

Um die Whiteboards vernetzen zu können, müssen diese und deren Benutzer eindeutig identifizierbar sein. Vor allem der Sonderfall, dass ein Whiteboard von mehreren Benutzern zeitgleich bearbeitet werden kann, muss beachtet werden.

Der Anmeldeprozess soll so einfach wie möglich sein. Der Benutzer kann lediglich einen Namen für das Whiteboard eingeben, unter dem es dann für die anderen Benutzer sichtbar ist. Um in großen Netzwerken gefunden werden zu können, können Whiteboards bestimmten Gruppen zugeordnet werden. So können z. B. Whiteboards in Länder oder auf Orte bezogene Gruppen eingeteilt werden, aber auch in logischen Gruppen, wie Firmen, Abteilungen oder Themen. Eine Mitgliedschaft in mehreren Gruppen sollte daher ebenfalls möglich sein. Abbildung 4.1 veranschaulicht diese Struktur.

Eine weitere Möglichkeit, den Login-Prozess besonders einfach zu gestalten wäre auch bereits vorhandene Anmelde-Systeme, wie z. B. die *Windows LiveId*, oder Protokolle, z. B. *AIM* oder *Skype* als Basis für den Anmeldeprozess zu verwenden. Alle diese proprietären Systeme bieten APIs an, um diese in eigenen Anwendungen verwenden zu können. Allerdings begibt man sich dabei in Abhängigkeit eines fremden Herstellers.

4.2.3 Sessions und Authentifizierung

Ein weiteres Konzept, welches umgesetzt werden soll, sind die verschiedenen Sessions. Eine Session wird dabei immer auf dem zentralen Server gespeichert und kann dabei von einem oder mehreren Whiteboards bearbeitet werden. Die Bearbeitung kann gleichzeitig oder auch hintereinander erfolgen. Erfolgt diese zeitgleich, so müssen die Benutzer gegenseitig erfahren, was der andere Benutzer gerade macht. Darauf wird auch im folgenden Abschnitt eingegangen.

Eine Session speichert dabei alle auf ihr ausgeführten Aktionen, sei es das Hinzufügen von Objekten, das Verschieben, oder auch das wieder Wegnehmen. Zu jeder Aktion soll auch gespeichert werden, wer diese ausgeführt hat.

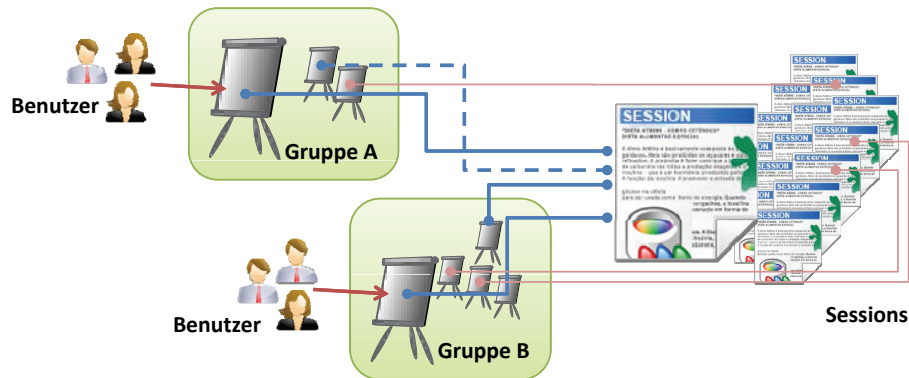


Abbildung 4.1: Konzept: Whiteboards in unterschiedlichen Gruppen ein-geordnet können an beliebigen Sessions teilnehmen.

Somit kann im Nachhinein ein Protokoll über alle durchgeführten Aktionen erstellt werden.

Des Weiteren soll es es auch möglich sein, an jeden beliebigen Punkt der Session zurückzukehren. Dabei kann man zwei Varianten wählen, eine destruktive und eine konstruktive. Bei der destruktiven Variante kann die Session zu einen beliebigen Punkt zurückgespult werden und von diesem Punkt ausgehend weiter bearbeitet werden. Alle Bearbeitungen, welche nach diesem Punkt geschehen sind, gehen somit verloren, ähnlich der Rückgängig-Funktionalität in den meisten Anwendungen. Bei der konstruktiven Variante kann ab einem bestimmten Punkt die Session kopiert werden, somit können beide Bearbeitungswege erhalten bleiben.

Ein Session darf natürlich nicht von allen Whiteboards automatisch ein-gesehen werden, daher ist eine Authentifizierung notwendig. Dabei ist es demjenigen, welcher die Session anlegt, möglich andere Benutzer bzw. Whiteboards zu einer Session einzuladen bzw. zu autorisieren. Dabei soll es natür-lich verschiedene Autorisierungsstufen, ähnlich wie bei einem Dateisystem, geben: Lesen, Lesen & Schreiben, sowie Administration. Lediglich der Admi-nistrator kann auch weitere Whiteboards für eine Session autorisieren oder diese wieder löschen.

Unterscheidung der Benutzer

Wie in *Instant-Messaging*-Anwendungen muss natürlich angezeigt werden, wer gerade mit der Session verbunden ist. Über eine Informationsanzeige können der Ersteller der Session bzw. eingeladene Whiteboards und Benut-zer eingesehen werden und Einstellungen getroffen werden. Sobald an einer Session etwas geändert wird, können optional der Besitzer oder interessierte Autorisierte eine Benachrichtigung erhalten.

Wenn mehrere Benutzer auf ein und demselben Whiteboard arbeiten, ist es notwendig diese einzeln zu identifizieren. Dies kann über die eindeutigen Adressen der digitalen Stifte geschehen (siehe Abschnitt 5.1.1). So kann zum Beispiel ein Benutzer seinen eigenen Stift mitbringen, und sobald er sich mit dem Whiteboard verbindet, kann seine Identität vom Server geladen werden. Seine Identität muss aber nicht nur aus einem Namen bestehen, sondern auch noch weitere Informationen, wie ein persönliches Profil oder ein Foto beinhalten.

Benutzer am Computer, welche mit Maus und Tastatur bzw. einem *StylusPen* arbeiten, registrieren ihre „Whiteboard“-Anwendung, und erhalten somit eine eindeutige Whiteboard-*ID*, welche dann ebenfalls zur Benutzer-Identifizierung verwendet werden kann.

4.2.4 Kollaborative Funktionalitäten

Kommunikation

Die Schwierigkeit dieser Applikation besteht darin, dass nicht nur mehrere Benutzer auf demselben Whiteboard arbeiten können, sondern auch auf mehreren verschiedenen Whiteboards über große Distanzen hinweg.

Wie erste Versuche gezeigt haben, ist eine reine grafische Verbindung (bei der man nur die Whiteboard Anwendung vor sich hat) nicht praktikabel. Arbeitet man auf einem Computer, so kann eine Text/Chat-Verbindung genügen, bei einer Installation, an der man keine Tastatur zur Verfügung hat, stellt sich eine Audio-Verbindung als geeignete Kommunikation dar. Die Integration einer bestehenden *VoIP*-Anwendung sollte daher angedacht werden.

Gerade wenn viele Benutzer vor einem Whiteboard arbeiten, wäre eine zusätzliche Videoverbindung wünschenswert. So kann nämlich eingesehen werden, wer oder wie viele sich gerade vor der Wand aufhalten und arbeiten. Genau die Möglichkeit zu sehen, wer gerade arbeitet, ist eine der großen Notwendigkeiten in dieser Anwendung. Die Aktivität der Benutzer sollte auf mehrere unterschiedliche Arten visualisiert werden.

Arbeitsvorgänge hervorheben

Da Benutzer und Objekte identifizierbar sind, ist es möglich Striche, welche soeben bearbeitet werden, zu markieren. Die Striche eines Benutzers können ähnlich wie mit einem Textmarker hervorgehoben werden.

Eine weitere Möglichkeit wäre auch ein kleines Fähnchen oder einen Stift an genau der Stelle anzuzeigen, an der sich der Benutzer gerade befindet. In diesem Fähnchen könnte nicht nur der Name angezeigt werden, sondern auch ein Avatar, ein Foto bzw. ein Live-Video der Person.

Auf jeden Fall muss zumindest in einer Liste, welche die Namen der Benutzer bzw. Whiteboards beinhaltet, auch der jeweilige Status, den die

Person hat, angezeigt werden. Neben der Anzeige, ob der Benutzer gerade an- oder abgemeldet ist, und der Visualisierung des soeben verwendeten Werkzeuges, sollte es auch eine Möglichkeit geben, Wünsche oder Erwartungen zu zeigen. So sollte man z. B. ein Symbol aktivieren können, welches den Anderen signalisiert, dass eine Antwort oder eine Hilfestellung erwünscht wäre.

Des Weiteren ist es sehr störend, nicht genau zu wissen, wo sich der andere Benutzer gerade befindet, da eine Position erst beim Zeichnen übertragen werden kann. So kann es passieren, dass man an der selben Stelle zu zeichnen beginnt. Ist dieses Problem an einer Wand insofern behoben, in dem man sich unweigerlich im Wege steht, so existiert diese natürliche Barriere bei verteilten Anwendungen nicht. Eine mögliche Lösung dieses Problems könnte wie in Abschnitt 3.1.1 beschrieben die Anzeige von Schatten sein. An einer digitalen Wand wäre es auch nicht allzu schwierig die Umrisse einer Person über ein kamerabasiertes Lokalisierungsverfahren herauszufinden. Gleichzeitig könnte eine Maus mit Hilfe des *Hover*¹-Status die aktuelle Position liefern. Leider liefert der digitale Stift, wie in Abschnitt 5.1.1 beschrieben, keinen *Hover*-Status zurück und es kann keine einheitliche Lösung dieses Problems für die Anwendung auf Desktop-Rechnern und auf einer digitalen Wand geben.

Neben den Symbolen, mit denen man den anderen Benutzern etwas mitteilen oder ausdrücken möchte, ist auch ein dem Zeigestab ähnliches Werkzeug notwendig. Auf einer Wand kann man sehr einfach mit der Hand an eine Stelle deuten, dieselbe Geste muss mit Hilfe des „Zeigewerkzeuges“ auch digital umgesetzt werden.

Multimediaelemente

Vor allem die Kombination von manuell angefertigten Skizzen und handschriftlichen Notizen, mit bereits vorbereiteten Bildern, Präsentationen und Dokumenten sollen den großen Vorteil gegenüber bereits vorhandenen *Sketching*-Anwendungen bringen. So soll es möglich sein, dieselbe Applikation zur Darstellung von bereits präsentierfähigen Inhalten zu verwenden, und gleichzeitig kann ein Teilnehmer auf diesen interagieren oder annotieren.

Entstehungsprotokoll

Neben der aktuellen Übersicht über die Aktivitäten der Benutzer sollte es auch im Nachhinein möglich sein dies nachzuverfolgen. Vor allem die Erstellung eines Entstehungsprotokolls wäre für viele Anwendungen wünschenswert. Auf der einen Seite kann es hilfreich sein, einfach eine Session im Nachhinein (in schneller Geschwindigkeit) abzuspielen um so den gesamten

¹Unter *Hover* bezeichnet man das Erkennen der Mausbewegung ohne dabei eine Maustaste zu drücken, oder den Stift auf die Oberfläche abzusenken.

kreativen Verlauf zu verfolgen, auf der anderen wäre aber auch eine interaktive Version des Protokolls wünschenswert.

Das gesamte Entstehungsprotokoll könnte in einer eigenen Anwendung interaktiv gestaltet werden. So sollte es möglich sein einen Benutzer auszuwählen und zu verfolgen, was er alles beigetragen hat. Oder man könnte einen Bereich am Whiteboard auswählen, und kann sich die daran beteiligten Teilnehmer anzeigen lassen.

Kapitel 5

Technologien

In diesem Abschnitt wird auf die grundlegenden Technologien und Techniken eingegangen, auf welchen die Implementierung des Prototypen basiert.

5.1 Interaktives Whiteboard

Als Hardwaresetup für den in dieser Arbeit beschriebenen Prototypen dient derselbe Aufbau wie für *Intoi* (siehe Abschnitt 3.2.3, bzw. [4]). Dabei wird ein digitaler Stift – ein *Anoto Pen*¹ – zur Interaktion verwendet.

¹www.anoto.com

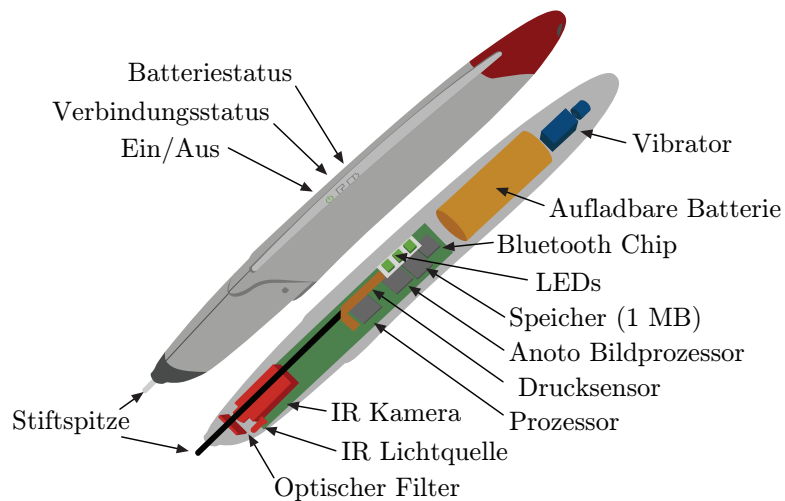


Abbildung 5.1: *Anoto Pen*: Neben einer Kugelschreibermine befinden sich auch eine Infrarot-Kamera in der Stiftspitze. (Grafik von Jakob Leitner)

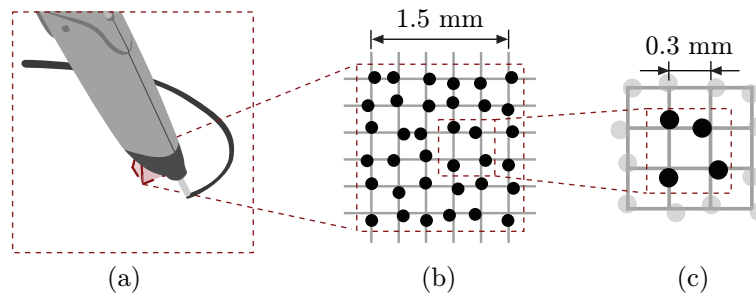


Abbildung 5.2: *Anoto Pattern*: Dieses Muster, welches auf Papier oder eine Projektionswand gedruckt werden kann, wird vom Stift für die Lokalisierung benötigt. (Grafik von Jakob Leitner)

5.1.1 Anoto Technologie

Für das Lokalisieren der digitalen Stifte wird die *Anoto Pen&Paper* Technologie verwendet. Abbildung 5.1 zeigt eine schematische Darstellung des Aufbaus eines Anoto Stiftes. Neben einer Kugelschreibermine (welche durch eine *Stylus*-Spitze² ersetzt werden kann) befinden sich auch eine Infrarot-Diode, sowie eine Infrarot-Kamera in der Spitze des Stiftes. Mit Hilfe der Kamera wird ein gedrucktes Punktmuster ausgelesen, welches in Abbildung 5.2 dargestellt ist.

Die Punkte sind grundsätzlich auf einem Raster angeordnet, werden aber in eine von vier verschiedenen Richtungen verschoben. Da die Kamera des Stiftes immer mindestens 6×6 Punkte erkennen muss, kann diese genau eine von 4^{36} möglichen Kombinationen erkennen. Der Abstand des Rasters beträgt 0,3 mm, somit kann eine Oberfläche von ca. 60 Millionen km^2 auf einen halben Millimeter eindeutig adressierbar gemacht werden.

Die Vorteile dieser Technologie liegen sicherlich in der Genauigkeit der Positionierung des Stiftes. Des Weiteren erkennt der Stift auch den Anpressdruck der Stiftspitze auf der Oberfläche. Ein Nachteil ist allerdings, dass das *Pattern* immer für den Stift sichtbar sein muss und ein Überdecken des Musters nicht möglich ist. Der Überdruck mit denen im Druck gebräuchlichen Farben *Cyan*, *Magenta* und *Yellow* (Gelb) ist möglich, lediglich darf keine schwarze (K) Farbe verwendet werden, da diese von der Infrarot-Kamera gelesen werden kann. Auch eine Projektion auf eine diffuse Oberfläche ist ohne Probleme möglich, solange kein direktes Licht die Kamera des Stiftes beeinträchtigt.

Die gesammelten Daten können am Stift gespeichert werden und später per *Bluetooth* oder über eine *Dockingstation* auf einen Computer übertragen werden. Es ist aber auch möglich, die Daten direkt zum Computer per *Blue-*

²Eine *Stylus*-Spitze bezeichnet eine weiche Plastik-Spitze, wie sie bei *PDA*s oder *Tablet-PC*s zum Einsatz kommt.

tooth zu übertragen. Jedes Datenpaket enthält neben der Position und des Anpressdruckes auch die eindeutige *ID* des Stiftes und der aktuellen Seite, sowie einen Zeitstempel.

Für den kommerziellen Vertrieb einer *Anoto* Lösung müssen Lizenzgebühren an *Anoto* bezahlt werden. Die Stifte werden von unterschiedlichen Herstellern angeboten, darunter *Destiny PLC*³, *Maxell*⁴ und *Nokia*⁵.

Weitere mögliche Tracking-Verfahren

Optisches Tracking Mit Hilfe von Kameras können z. B. Finger, Hände, Stifte und andere physikalische Objekte lokalisiert werden (vgl. [29], [7], sowie *SMARTBoard*⁶(DVit)).

Elektromagnetisches Tracking wird vorwiegend in Grafik-Tablets und Tablet-PCs eingesetzt.

Ultraschall Tracking Die Stifte (z. B. *Mimio*⁷) senden Ultraschallsignale aus, welche von einer Empfängerleiste aufgenommen werden. Probleme entstehen wenn sich Objekte zwischen der Empfängerleiste und den Stiften befinden.

Analog Resistives Tracking Bei diesem Verfahren befindet sich zwischen zwei elektrisch leitenden Folien ein dünner Luftspalt. Sobald jedoch die Oberfläche berührt wird, wird der Kontakt geschlossen und die Position kann berechnet werden. Dieses Verfahren funktioniert nur mit *einem* Stift oder Finger und mehrere Benutzer sind daher nicht möglich.

5.1.2 Rückprojektionsaufbau

Ein möglicher Aufbau des digitalen Flipcharts ist in Abbildung 5.3 zu sehen. Mit diesem speziellen Setup ist es möglich von der Rückseite der Projektionswand zu projizieren. Das *Anoto Pattern*, welches auf eine *BackLight*⁸ Folie gedruckt wird, ist dabei in einem Alurahmen eingespannt und wird mit zwei Acrylplatten geschützt.

³*Destiny IO2* (früher *Logitech*), www.destinyplc.co.uk

⁴*Maxell PenIt DP-201*, www.maxell.co.jp/e/products/industrial/digitalpen/

⁵*Nokia SU-27W*, <http://europe.nokia.com/su-27w>

⁶*SMARTTech*, www.smarttech.com

⁷*Mimio*, www.mimio.com

⁸Diese diffuse Folie ist zu einem gewissen Teil lichtdurchlässig und wird daher z. B. für Litfaßsäulenwerbung verwendet.



Abbildung 5.3: Rückprojektionsaufbau. (Grafik von Verena Lugmayr)

Eine Projektion von vorne ist ebenso möglich, wird allerdings kein Kurzstreckenprojektor⁹ verwendet, so beeinträchtigt die Schattenbildung die Arbeit des Benutzers erheblich.

Dieser Aufbau ermöglicht es zusätzliche Geräte zu montieren, z. B. Infrarot-Beleuchtung und Kameras für diverse andere Trackingverfahren (z. B. Hand-Tracking).

Direkte versus Indirekte Eingabe

Mit Hilfe dieses Setups ist eine direkte Interaktion auf der Arbeitsfläche (direkte Eingabe) möglich. Darunter versteht man, dass die Objekte an Ort und Stelle bewegt oder verändert werden. Im Gegensatz dazu wird bei einer klassischen Computerbedienung mit einer Maus lediglich die indirekte Eingabe verwendet, da die absoluten Bewegungen der Maus nicht der tatsächlichen Bewegung des Maus-Cursors am Bildschirm entsprechen.

Diese direkte Eingabe hat auf der einen Seite viele Vorteile, da der Benutzer mit dem Finger oder Stift Objekte auf der Stelle anfassen oder bearbeiten kann, auf der anderen Seite bringt diese Eingabemethode auch Probleme mit sich. Zum Beispiel bei Menüs muss man darauf achten, dass wichtige Bereiche nicht von der Hand oder von einem physikalischen Objekt verdeckt werden. Auf größeren interaktiven Flächen kann es für den Benutzer störend sein, wenn er ein Bild vom linken zum rechten Bereich verschieben möchte und

⁹Ein Kurzstreckenprojektor besitzt spezielle Linsen, um nur eine kurze Distanz zur Leinwand zu benötigen. Dadurch wird das Bild in einem besonders steilen Winkel projiziert.

dabei mehrere Meter zurücklegen muss. Dazu siehe auch [6].

5.2 Softwaretechnologien

5.2.1 *.NET Framework 3.0*

Neben der *Windows Presentation Foundation* (*WPF*, siehe Abschnitt 5.2.2) beinhaltet das von Microsoft 2006 herausgegebenen *.NET Framework 3.0* auch noch die *Windows Communication Foundation* (*WCF*), die *Windows Workflow Foundation* (*WWF*, zur Beschreibung und Umsetzung automatisierter und regelbasierter Datenflüsse), sowie *Windows CardSpace*.

WCF ist ein dienstorientiertes Nachrichtensystem für verteilte Anwendungen. Durch die *WCF* werden unterschiedlichste Kommunikationstechnologien, z. B. TCP/IP, WebServices, DCOM¹⁰, Enterprise Services, zu einer einzigen Programmierschnittstelle zusammengefasst. Dazu wird in der *WCF* ein verteilter Endpunkt durch die Trennung von Adresse (URI, Ort des Dienstes), *Binding* (Anbindung, Art des Protokolls und der Kodierung), sowie *Contract* (Vertrag/Dienstdefinition, verfügbare Methoden) abstrahiert.

Mit Hilfe von *CardSpace*¹¹ wird die Authentifizierung auf Webseiten und Online-Diensten durch eine am Client installierte Software, welche neben *Windows* auch für *Apple* und *Linux* verfügbar ist, erledigt. Diese Software soll diese Authentifizierung nicht nur einfacher, sondern vor allem sicherer und vertrauenswürdiger machen. Dazu wird das „Kreditkartenprinzip“ angewendet, bei dem der Benutzer verschiedene Karten in einer Geldbörse aufbewahrt. *CardSpace* verwaltet diese Karten und bietet Sie dem Benutzer an, sobald eine *CardSpace* Authentifizierung bei einer Applikation oder Webseite nötig ist.

.NET Framework 3.5 & Silverlight

Silverlight ist insofern sehr interessant, da man dadurch schon bestehende Applikationen, Benutzerschnittstellen und den dahinterliegenden *.NET/C#* Code sehr einfach ins Web portieren kann. Eine *Silverlight* Oberfläche wird mit Hilfe von XAML (siehe Abschnitt 5.2.2) beschrieben und kann neben *C#* Code aber auch durch *JavaScript* Code gesteuert werden. Die Bedienoberfläche kann des Weiteren dynamisch zur Laufzeit geändert werden und setzt dabei auf in AJAX übliche XML Schnittstellen auf. Obwohl *Silverlight* mit *Adobe Flash/Flex*, *JavaFX* und *Apple Quicktime* konkurriert, will es sich besonders durch seine Multimedia- bzw. Video- und Audiofähigkeiten durchsetzen.

¹⁰ *Distributed Component Object Model* ist ein objektorientiertes RPC-System. Es wurde von Microsoft definiert, um die Technologie COM über ein Netzwerk kommunizieren zu lassen.

¹¹ <http://netfx3.com/content/WindowsCardspaceHome.aspx>

Programm 5.1: „Hello World!“ in XAML.

```
1 <Window x:Class="wpfDemo.HelloWorldWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     Title="Hello World" Height="300" Width="300" Background="#FFFFFFFF">
5     <StackPanel>
6         <TextBlock Text="Hello World!" FontSize="36" FontWeight="Bold"
7             Foreground="#00000000" />
8     </StackPanel>
9 </Window>
```

Das *.NET Framework 3.5*, welches Ende 2007 publiziert wurde, bietet darüber hinaus mit *LINQ*¹² eine einfache Abfragesprache für Datenbanken und *Collections*. Das besondere daran ist, dass schon zur Kompilierzeit die Abfragesyntax auf mögliche Fehler überprüft werden kann.

5.2.2 Windows Presentation Foundation

Die *Windows Presentation Foundation (WPF)* [18] ist eine neuartige GUI bzw. Präsentations-Programmierschnittstelle für *Microsoft Windows*. *WPF* funktioniert auf *Windows XP* und *Server 2003*, wurde aber vor allem für *Windows Vista* entwickelt. Bereits einige Dienstprogramme in *Windows Vista* wurden mit der *WPF* implementiert, daher ist *WPF* fest in diesem Betriebssystem verankert.

XAML

XAML (eXtensible Application Markup Language) ist eine auf XML basierende Beschreibungssprache für grafische Benutzeroberflächen (für *WPF*) und Workflows (*WWF*). *XAML* ist mit der von *Adobe* für *Flex* bzw. *Flash* entwickelten Beschreibungssprache *MXML* vergleichbar. *XAML* beschreibt nicht nur das User Interface selbst, sondern auch eigene Steuerelemente (hinsichtlich Gestaltung und Verhalten), Datenflüsse und Datenbindungen, Ereignisse und andere Eigenschaften.

Programm 5.1 zeigt die Beschreibung eines Fenster in *XAML*, in welchem „Hello World!“ angezeigt wird. Das verwendete *StackPanel* (ein Layout-Container) ordnet Elemente vertikal oder horizontal nacheinander an. Bei nur einem Element wird dieses in der Mitte ausgerichtet.

¹²Language Integrated Query

LAYOUTS, VEKTORGRAFIKEN & ANIMATIONEN

In XAML können nämlich auch komplette Vektorgrafiken beschrieben werden (siehe dazu [18, Kapitel 4]). Dadurch ist es nicht nur möglich vollständige auflösungsunabhängige User Interfaces zu schaffen, sondern auch die Größe der Grafikdateien wird ohne Qualitätsverlust auf das Geringste reduziert. Dabei lehnt sich XAML an den schon bestehenden SVG Standard (vgl. [28]) an. Mittlerweile werden für bereits alle gängigen Grafik- und Design-Programme XAML-Exporter angeboten.

Mit Hilfe von Layout-Containern können die Steuerelemente so platziert werden, dass sie in unterschiedlichsten Auflösungen richtig angezeigt werden. Da alle grafischen Elemente in WPF als Vektorgrafik importiert oder beschrieben werden, ist ein verlustfreies Vergrößern oder Verkleinern schon vorhandener Elemente problemlos möglich. Somit kann nicht nur die Position auflösungsunabhängig gestaltet sein, sondern auch die Größe eines Objektes kann ohne Verluste dynamisch gestaltet sein.

Animationen können wie in *Adobe Flash* mit Hilfe einer Zeitleiste gestaltet werden. Der große Unterschied zwischen *Flash* und XAML/WPF besteht allerdings darin, dass *Flash* bildbasiert abläuft. Das bedeutet, dass nach einer bestimmten Zeit ein neues Bild (*Frame*) angezeigt bzw. die `onEnterFrame()` Methode abgearbeitet wird. Dabei wird das gesamte Fenster (oder der betroffene Abschnitt) neu gezeichnet. In WPF gibt es keine ständige `render()` Methode die kontinuierlich aufgerufen wird, sondern ein Element, welches wieder gezeichnet werden soll, muss dies melden. Darum werden die einzelnen Animationsschritte in WPF auch über einen *Timer* berechnet und dann neu gezeichnet.

Stile

Mit der Definition von Stilen [18, Seite 307ff.] können schon vorhandene *User Controls* (zB. `Label`, `TextBox`, `Button`, ...) neu definiert bzw. gestaltet werden. Nicht nur das Aussehen kann hierbei geändert werden, sondern auch das Verhalten. So kann z. B. ein *Label* auf einen Mausklick anders reagieren, als das ursprüngliche Element.

Datenbindungen

Datenbindungen (*Data Binding* [18, Seite 259ff.]) machen es möglich Eigenschaften eines WPF Elements auf ein Anderes weiterzugeben. Eine Datenbindung sieht wie folgt aus:

```
1 <Label Content="{Binding Path=Value, ElementName=s1, Mode=Default}"/>
2 <Slider Name="s1" Value="50" Minimum="0" Maximum="100" />
```

Neben der Möglichkeit von einem Element auf ein anderes zu verweisen, kann man aber auch ganze Objekte bzw. *Collections* auf ein `UIElement` binden. Wenn man z. B. einem Anwendungsbereich ein `Stroke` Objekt zuweist,



Abbildung 5.4: Mit dem *MultiPoint SDK* ist es möglich in Applikationen mehrere Mäuse und Tastaturen per *ID* unterscheiden zu können.

können Eingabefelder die Eigenschaften dieses **Stroke** Objektes übernehmen (Farbe, Strichstärke, ...).

Entwicklung eigener Steuerelemente

Neben der Möglichkeit mit Hilfe von Stilen schon vorhandene Elemente neu zu gestalten, kann man auch gänzlich eigene Steuerelemente entwerfen. Dies ist vor allem dann nötig, wenn man ein Steuerelement benötigt, welches neue Eigenschaften oder ein neues Verhalten mitbringen soll.

Ein Beispiel für ein eigenes Steuerelement findet man in Kapitel 6.4, in dem es notwendig war einen Container zu schaffen der unter anderem Strich- und Vektordaten, sowie Bildelemente auf einer grenzenlosen Zeichenfläche verwaltet.

Bei der Erstellung eines eigenen Steuerelements gibt es verschiedene Möglichkeiten diese von Oberklassen abzuleiten. **UIElement** ist die Grundklasse aller Elemente, welche sich in einem User Interface befinden können. Erbt man aber zum Beispiel die Eigenschaften eines **Panel**s so kann man einen eigenen *Layout-Container* definieren. Dabei müssen die Positionen und Größen der einzelnen im Container gelagerten Elemente bestimmt werden.

5.2.3 Verwaltung mehrerer Eingabegeräte

Werden an einem Computer mehrere Mäuse und Tastaturen angeschlossen, so wird aber dennoch nur ein Mauszeiger oder ein Eingabefeld bedient. Mit Hilfe des *Microsoft Windows MultiPoint SDK* [13] wird es allerdings möglich in *.NET/WPF* Anwendungen mehrere Maus-Zeiger zu unterstützen. In [20]

wird beschrieben, wie so vor allem für Kinder und Schulen in Entwicklungsländern Applikationen entwickelt werden können, mit denen mehrere Kinder gleichzeitig und gemeinsam an einem Computer arbeiten können (siehe Abb. 5.4).

Beim Einsatz des *Multipoint SDKs* können allerdings keine Standard-Steuerelemente verwendet werden. Durch Ableitung von einem Button und einer Schnittstelle aus dem *Multipoint SDK* können aber die meisten gängigen Elemente mehrbenutzerfähig gemacht werden. Im Grunde werden neben dem Maus- und Tastatur-Ereignissen auch *Hardware IDs* übertragen.

5.3 Netzwerkprogrammierung

Um Whiteboards zu vernetzen, wird (wie in Abschnitt 6.3 beschrieben) ein eigenes Nachrichtenprotokoll entwickelt, welches auf dem *Transmission Control Protocol* basieren wird.

5.3.1 *Transmission Control Protocol* (TCP)

Bei der Übertragung dürfen die Nachrichten nicht verloren gehen, in der selben Reihenfolge und nicht mehrfach ankommen, und auch nicht durch etwaige Fehler verändert werden. Um dies zu gewährleisten, wird das *Transmission Control Protocol* (kurz: TCP) als Kommunikationsprotokoll verwendet. Folgende Eigenschaften von TCP begünstigen diese Anforderungen, dazu siehe auch [11, Seite 238f] bzw. [25].

Verbindungsmanagement Ein Verbindungs-Management in 3 Phasen ermöglicht einen gesicherten Verbindungsaufbau (*Three Way Handshake*), die Aufrechterhaltung der Verbindung während der gesamten Datentransferphase und einen gesicherten Abbau der Verbindung.

Flusskontrolle und Fehlerbehandlung Das TCP-Protokoll gewährleistet, dass die Daten korrekt übertragen werden, indem jedes Datenpaket quittiert werden muss. Eine fortlaufende Nummerierung aller übertragenen Datenbytes (Sequenznummern), die Bestätigung der empfangenen Daten (*Acknowledgements*) und ein Fenstermechanismus verhindern den Verlust von Daten und den Überlauf von Datenpuffern. Das TCP-Protokoll überprüft die empfangenen Daten auf Fehler und meldet diese an höhere Protokolle weiter oder fordert das fehlerhafte Datensegment vom Sender erneut an. Übertragene Datensegmente müssen nach einem definierten Zeitraum vom Empfänger quittiert werden. Findet nach dem Ablauf dieser Zeit keine Bestätigung für den Empfang der Daten statt, so werden die Daten erneut übermittelt.

Multiplexen von Verbindungen Um mehreren Prozessen den gleichzeitigen Zugriff auf TCP zu ermöglichen, werden mehrere *Ports* (Zugangsadressen) zur Verfügung gestellt. Ein *Socket* (Verbindungspunkt) bildet sich aus einer IP-Adresse und der Port-Adresse.

Kapitel 6

Implementierung

Die Implementierung des Prototypen erfolgt in C# mit dem *.NET Framework 3.5*. Für die grafische Benutzeroberfläche wird die *Windows Presentation Foundation* (siehe Kapitel 5.2.2) verwendet. Der Prototyp basiert in einigen Punkten auf das in Abschnitt 3.2.3 beschriebene Projekt *Intoi*.

Warum ist ein neuer Prototyp notwendig?

Da *Intoi* als Anwendung konzipiert wurde, welche lediglich auf *einem* Hardware-Setup läuft, aber von mehreren Benutzern bedient werden kann, wurde bei der Implementierung auf ein Kommandosystem [8, Abschnitt 5.2] verzichtet. Die einzelnen *Pen* Objekte, welche jeden Benutzer/Stift kapseln und die einzelnen Werkzeuge verwalten, können direkt auf die Datenstruktur zugreifen, welche in einem zweiten *Thread* auf den Bildschirm gerendert wird.

Dies hat zwar einen Leistungsvorteil, aber macht es nur sehr schwer möglich eine zusätzliche Schicht mit Kommandos oder ähnlichem einzuführen, um ein sicheres verteiltes Arbeiten zu ermöglichen. Da es mit der WPF einfach möglich ist, Benutzeroberflächen zu gestalten, die nicht dem *WIMP*-Standard entsprechen, bietet sich diese API für diesen neuen Prototypen an. Durch die einfach konfigurierbare XAML Sprache soll das Ausprobieren verschiedener Funktionen im Prototypen erleichtert werden.

6.1 Anwendungsarchitektur

Abbildung 6.1 zeigt die grundlegenden Teile der gesamten Whiteboard Anwendung. Dabei bauen zwei Programmbibliotheken direkt am *.NET Framework* auf, sind aber von der Benutzeroberfläche unabhängig. Die Kernprogrammbibliothek (*Whiteboard.Core*) beinhaltet alle Manager-Klassen, sowie die gemeinsamen Kommandos und Ereignisse, sowie deren Verwaltung. Die Netzwerkbibliothek (*Whiteboard.Network*) implementiert das in Abschnitt 6.3 vorgestellte Nachrichtensystem, sowie die serverseitige Session-Verwaltung.

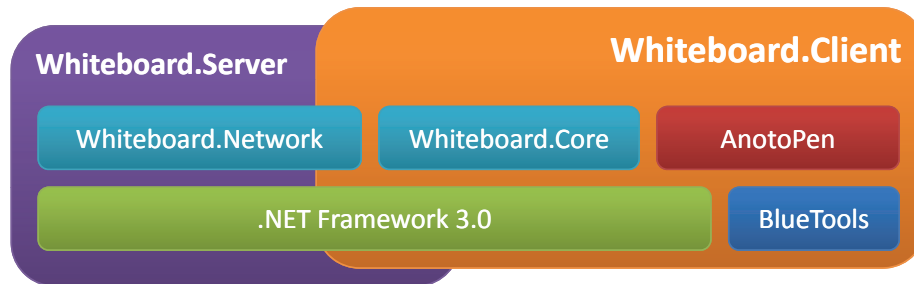


Abbildung 6.1: Struktur der Anwendung.

Der Whiteboard Server kann als *Windows* Systemdienst¹ installiert werden. Der Port muss dabei im Service-Manager als Startparameter übergeben werden. Wird kein Port übergeben, so läuft der Whiteboard Server standardmäßig auf Port 15000. Der Server kann auch als Konsolenanwendung zur Ausgabe von *Debug*-Meldungen gestartet werden.

Das Client-*Assembly*² (*Whiteboard.Client*, bzw. *Whiteboard.exe*) beinhaltet die erstellten GUI-Komponenten sowie die Programmlogik für den Client. Ebenfalls wird die selbst erstellte *AnotoPen* Bibliothek benötigt um mit den digitalen Stiften kommunizieren zu können. Diese Bibliothek basiert auf der kommerziell erhältlichen *BlueTools*³ Bibliothek für .NET, welche einen einfachen Zugriff auf den *Microsoft* sowie den *Widcomm Bluetooth Stack*⁴ bietet.

6.1.1 Client/Server Architektur

Vor der Entscheidung über die Netzwerkarchitektur wurden mehrere kleine Prototypen entwickelt, mit denen das Verhalten getestet wurde.

Peer-to-Peer vs. Client/Server

Wie schon in Abschnitt 5.3 diskutiert, gibt es verschiedene Möglichkeiten bei der Wahl der Netzwerk- bzw. Anwendungsarchitektur. Grundsätzlich wäre eine *Peer-to-Peer (P2P)* Lösung sehr wünschenswert gewesen, da so die einzelnen Ereignisse auf dem direktesten Weg verteilt werden könnten. Allerdings bringt eine *P2P* Architektur auch einige Probleme mit sich: Es kann

¹Ein Systemdienst ist ein Programm, das automatisch beim Hochfahren von *Windows* startet und so lange im Hintergrund läuft, bis *Windows* heruntergefahren wird. Im Konzept ist er dem *Unix-Daemon* sehr ähnlich.

²Ein .NET Assembly ist ein bereits in die *Common Intermediate Language (CIL)* übersetzter Code (Bytecode), der von der *Common Language Runtime (CLR)* ausgeführt werden kann. Es gibt ausführbare Assemblys (.exe) und Programmbibliotheken (.dll).

³*Franson BlueTools*, www.franson.com/bluetooth/

⁴www.broadcomm.com

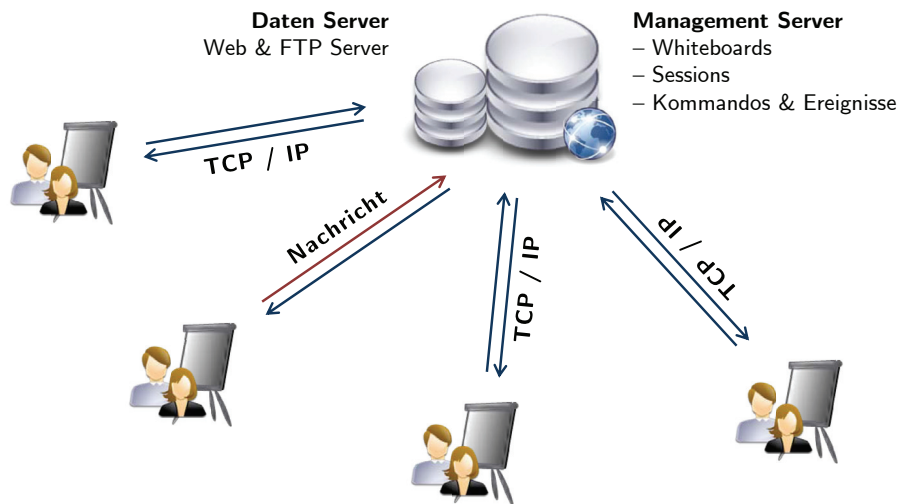


Abbildung 6.2: Neben dem Whiteboard Server, welcher Sessions und Kommandos verwaltet, kümmert sich ein Web- bzw. FTP Server um den Datei Up- und Download.

nur sehr schwer gewährleistet werden, dass die Ereignisse auf allen Rechnern gleichzeitig und in richtiger Reihenfolge ankommen, eine Inkonsistenz des Datenmodell ist daher sehr schnell wahrscheinlich.

Des Weiteren ist eine der Anforderungen, dass es eine zentrale Stelle geben soll, auf der alle Sessions verwaltet und gespeichert werden. Zuletzt gilt es noch zu Beachten, dass es auch notwendig ist, neben den „Echtzeit“-Ereignissen (z. B. Kommandos) auch große Datenmengen, z. B. Bilder oder Videos, zu übertragen, was aufgrund der heutigen Netzwerkkapazitäten nicht in Echtzeit geschehen kann.

P2P oder auch multiple Servereinheiten stellen gerade in Firmennetzwerken Probleme dar, da es dabei meist notwendig ist, direkte Verbindungen zu den Rechnern auf unterschiedlichen Ports herzustellen. Dies wird meist durch *Firewalls* aus Sicherheitsgründen verhindert.

Client, Server & Webserver

Abbildung 6.2 zeigt die nun gewählte Netzwerkarchitektur. Es wurde entschieden, eine klassische Client/Server-Architektur zu verwenden, bei der alle Ereignisse und Steuerbefehle über einen zentralen Server laufen, der diese validiert, in sein Modell einträgt und an alle Clients verteilt. Wie in der Grafik zu erkennen ist, besteht der Server aus zwei Teilen, dem Whiteboard Server und einem Web/FTP-Server, welcher zum Datenaustausch von Bildern, Videos, etc. verwendet wird. Auf die Implementierung der Netzwerkkommunikation wird in Abschnitt 6.3 näher eingegangen.

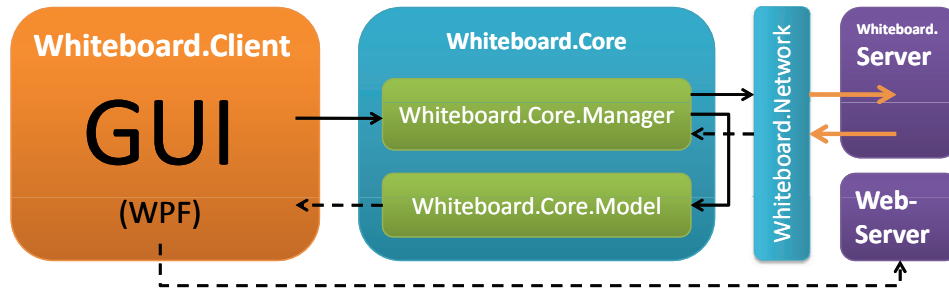


Abbildung 6.3: Die *Model-View-Controller* Architektur innerhalb der Whiteboard-Anwendung.

6.1.2 Model-View-Controller Architektur

Aus verschiedenen Gründen ist es notwendig, dass das Datenmodell sehr strikt von der Präsentationsschicht getrennt wird. Der Hauptgrund ist sicherlich, dass das Modell so klein wie möglich sein sollte, um es problemlos und schnell über das Netzwerk verteilen zu können (siehe Abschnitt 6.2). Auf der anderen Seite wurde auch angedacht, diese Applikation auf verschiedenen Plattformen und Geräten laufen lassen zu können, etwa auch auf *PDA*s oder *SmartPhones*. Da es dort andere Voraussetzungen für Eingabemöglichkeiten und Bildschirmausgabe gibt, würde lediglich das Datenmodell dasselbe bleiben.

Abbildung 6.3 zeigt, wie diese Architektur innerhalb der Whiteboard Anwendung umgesetzt wurde. Strichlierte Pfeile zeigen eine lose Bindung, z. B. über Ereignisse, an. Hier wird deutlich, dass die Netzwerkschicht mit dem Manager lediglich lose kommuniziert, da diese entbehrlich ist, wenn keine Verbindung zum Internet besteht. Die GUI kennt diesen Manager bzw. Controller und übergibt diesem z. B. Maus- oder Tastatur-Ereignisse. Der Controller kann entscheiden, ob dieses Ereignis über das Netzwerk abgehandelt werden muss oder direkt in des Modell eingetragen wird.

6.1.3 Kommandos und Ereignisse

Der Controller besteht grundsätzlich aus zwei Managern. Der Modell-Manager wird benötigt um eine Verbindung zum Wurzelement in der GUI herstellen zu können. Des Weiteren bietet dieser Hilfsfunktionen an, um die Elemente anhand einer *ID* (siehe Abschnitt 6.2.1) zu erkennen und bearbeitbar zu machen.

Neben dem Modell-Manager gibt es, wie in Abbildung 6.4 ersichtlich, den Kommando-Manager. Dieser ermöglicht nicht nur eine Rückgängig/Wiederholen (*Undo/Redo*) Funktionalität, sondern auch das Absetzen und Ausführen bestimmter Aktionen lokal oder über das Netzwerk. Die speziellen

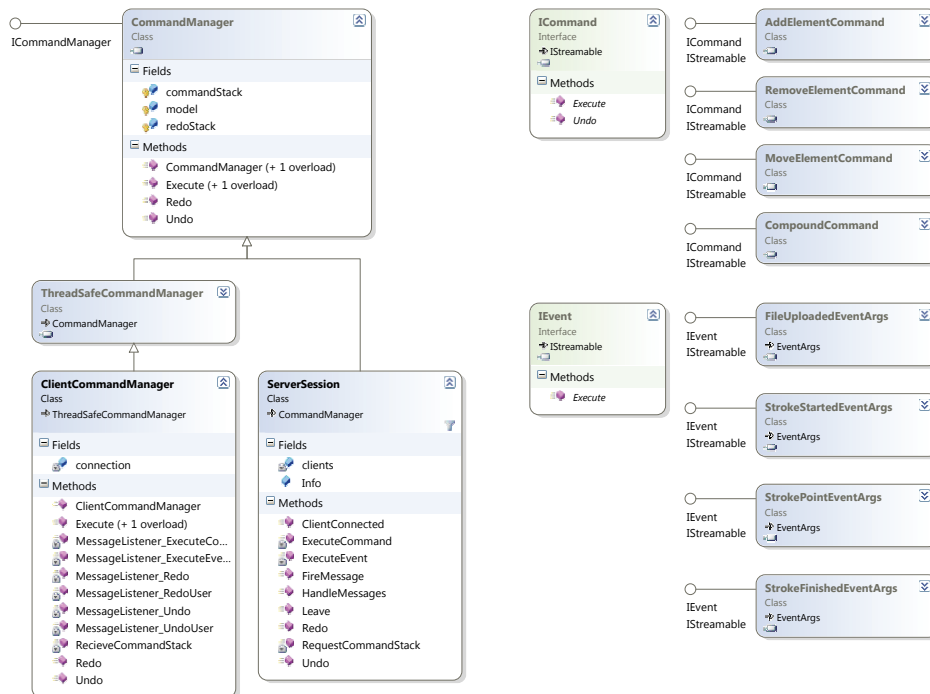


Abbildung 6.4: Der Kommando-Manager kann die aufgeführten Kommandos bzw. Ereignisse entweder lokal oder über ein Netzwerk behandeln.

Klassen `ClientCommandManager` bzw. `ServerSession` fungieren dabei als Proxys (siehe [8, Abschnitt 4.7]) und können zur Laufzeit gegen einen lokalen Kommando-Manager ausgetauscht werden.

Der Unterschied zwischen Kommandos und Ereignissen besteht darin, dass ein Ereignis zwar auf das Modell ausgeführt wird und dort auch gespeichert wird, allerdings nicht mehr rückgängig gemacht werden kann. Ein ausgeführtes Kommando hingegen wird am Kommando-Stack abgelegt und kann wieder reversiert werden. So wird z. B. das Hinzufügen eines Striches als Kommando abgesetzt und am Stack abgelegt. Die einzelnen Punkte eines Striches aber werden als separate Ereignisse über den Server verteilt. Der gesamte Strich kann nun wieder rückgängig gemacht werden, dabei sind aber alle einzeln übertragenen Punkte (um das Schreiben zu simulieren) im Strich-Modell gespeichert.

Auch sobald eine Datei auf den Webserver geladen wurde und die URL verfügbar ist, wird ein Ereignis ausgeführt. Dabei wird das `FileModel` (siehe nächster Abschnitt) Objekt derart verändert, dass es bei erneutem Anfordern des Kommando-Stacks die richtigen Informationen beinhaltet.

Programm 6.1: Die *ModelBase* Klasse, von der alle Modellelemente abgeleitet werden.

```

1 public abstract class ModelBase : IStreamable,
2     ISerializable, IIdentifiable
3 {
4     public event EventHandler<EventArgs> TransformationChanged;
5
6     protected Identifier id;
7     protected Matrix transformation;
8
9     public ModelBase()
10    {
11        id = Identifier.CreateUID();
12    }
13
14    public Identifier Id
15    {
16        get { return id; }
17        set { id = value; }
18    }
19
20    public Matrix Transformation
21    {
22        get { return transformation; }
23        set { transformation = value;
24            if (TransformationChanged != null)
25                TransformationChanged(this, new EventArgs());
26        }
27    }
28 }

```

6.2 Datenmodell

Um den Datenaustausch über das Netzwerk (siehe dazu Abschnitt 6.3) möglichst gering zu halten, wurde ein sehr einfaches Modell erstellt welches nur die wichtigsten Informationen zu den einzelnen Elementen enthält. Der Programmcode 6.1 zeigt Teile der *ModelBase* Klasse, von welcher jedes Modellelement abgeleitet werden muss. Abbildung 6.5 zeigt die Ableitungshierarchie bzw. alle Elemente des Datenmodells.

Jedes Modell implementiert folgende Schnittstellen: *IStreamable*, *ISerializable* und *IIdentifiable*. *IStreamable* ist eine eigene Schnittstelle, welche ein sehr einfaches aber komprimiertes Serialisierungsverfahren⁵ für das Netzwerkprotokoll zur Verfügung stellt. Auf dieses wird im Abschnitt 6.3 eingegangen.

ISerializable ist die Serialisierungsschnittstelle aus dem *.NET Framework*.

⁵Unter Serialisieren versteht man das Umwandeln eines Objektes in einen Bytestrom.

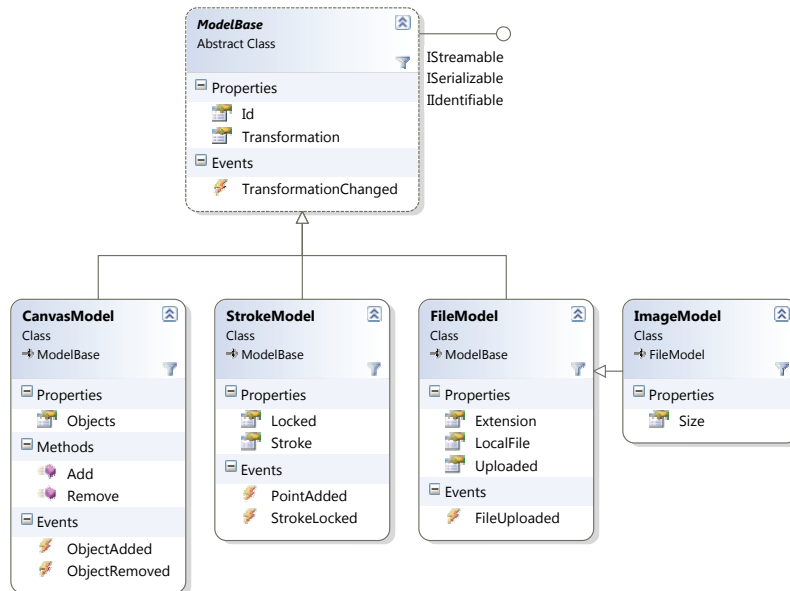


Abbildung 6.5: Diese Klassen des Datenmodells beschreiben alle Element am Whiteboard, sowie deren Eigenschaften.

work. Dieses kann zwar jegliche Art komplexer und auch generischer Datenobjekte serialisieren, dadurch werden aber zu viele Metadaten an den Datenstrom angehängt. Daher ist es für die Übertragung der Pakete über das Netzwerk nur sehr schlecht geeignet, da z. B. jedes serialisierte Objekt einen Präfix von ca. 128 Bytes bekommt. Diese Serialisierungsfunktion wird allerdings zum Speichern des Modells auf das Dateisystem verwendet.

Neben den Schnittstellen hat jedes Modell-Objekt auch zwei Eigenschaften, und zwar eine Transformationsmatrix, welche die Position und die Rotation des Objektes auf der Zeichenfläche bestimmt, sowie eine eindeutige Identität.

6.2.1 Identifizierbare Objekte

Zu guter Letzt implementiert das Modell die Schnittstelle `IIdentifiable`. Diese sorgt dafür dass jedes Element eine eindeutige Identität (*ID*) besitzt. Dazu gibt es ein eigenes `Identifier` Objekt. Dieses ist nötig, da es für Benutzer ohne eine Rückfrage auf den Server sofort möglich sein sollte ein Objekt zu erstellen. Des Weiteren sollte es möglich sein bestimmen zu können von welchem Benutzer, zu welcher Zeit, welches Element erstellt wurde. Außerdem gilt es zu beachten, dass auf einem Server mehrere *Sessions* (vgl.

Abschnitt 4.2.3) parallel laufen. Ein `Identifier` Objekt besteht aus folgenden Teilen:

- `UInt32 SessionId`
- `UInt32 UserId`
- `UInt32 ObjectId` (Zeitstempel)

Die Verteilung der Anwendung über ein Netzwerk macht diese eindeutige Identifizierbarkeit der Objekt ebenfalls notwendig. Neben den schon genannten Vorteilen, ist es mit Hilfe der *IDs* möglich, auf jeder laufenden Instanz der Applikation die Objekte zu identifizieren. Denn jede Instanz legt eigene Instanzen der Objekte an, es gibt daher keine andere Gemeinsamkeit (wie z. B. eine Speicheradresse) außer dieser *ID*.

6.2.2 Ereignisse bei einer Modell-Änderung

Da das Modell nicht nur am Client-Computer visualisiert werden soll, sondern auch am Server mitgespeichert und verändert werden soll, ist eine nur sehr lose Bindung zum eigentlichen Anzeige-Element notwendig. Dies wurde über sogenannte Ereignisse realisiert.

Jedes Objekt in .NET kann bestimmte Ereignisse anbieten [10, Kapitel 23], auf welches sich andere Objekte registrieren können. In der `ModelBase` Klasse wird z. B. das `TransformationChanged` Ereignis angeboten. Ein interessiertes anderes Objekt kann sich mit folgender Zeile auf dieses Ereignis registrieren:

```
1 model.TransformationChanged += new EventHandler(transformationChanged);
```

Dabei wird die `transformationChanged()` Methode des interessierten Objektes, ähnlich dem *Listener* Prinzip (vgl. [8, Abschnitt 5.7]), aufgerufen. Sodann kann die Ansicht aktualisiert und neu gerendert werden.

Der Vorteil dieses Systems ist, dass nicht jedes Mal das gesamte Modell durchlaufen werden muss, wenn es eine Änderung gibt. Das betroffene Objekt oder die Ansicht kann so gezielt aktualisiert werden.

6.3 Netzwerkprotokoll und Serveranwendung

Das Kernstück des gesamten Whiteboard Systems ist die Serveranwendung, welche erst die Verbindung zwischen den einzelnen Client-Anwendungen ermöglicht. Die dazugehörige Netzwerk-Architektur wurde bereits in Abschnitt 6.1.1 erläutert.

6.3.1 Wahl des Übertragungsprotokolls

Zu Beginn der Implementierung stand man vor dem Problem die Stift- und Mausereignisse möglichst effizient und schnell auf die einzelnen Client-Anwendungen zu verteilen. Wie schon in Abschnitt 6.1.1 erläutert, wäre hierbei eine *Peer-to-Peer* Verbindung, bzw. auch eine Übertragung über UDP eine Möglichkeit gewesen [25, Seite 211].

In einem Prototypen wurde UDP zur Verteilung der einzelnen Stifterereignisse (ca. 25-50 Ereignisse pro Sekunde, je ca. 48 Bytes) innerhalb eines geschlossenen Netzwerkes (alle Rechner waren im selben Subnetz) verwendet. Dabei war die Latenz (auch über drahtlose Verbindungen) vernachlässigbar. Bei Sendungen an entferntere Rechner über das Internet war nicht nur die steigende Latenz und die verlorenen Pakete sondern auch die nicht garantierte Reihenfolge der ankommenden Pakete ein zu großes Problem, was den Umstieg auf TCP veranlasste.

Da die Konsistenz zwischen den einzelnen Client-Anwendungen und dem Server aber gewährleistet sein muss, wird auf die sichere Verbindung über TCP vertraut. Vor allem der Verbindungsaufbau ist bei TCP um vieles langsamer als bei UDP, wo es ja praktisch keinen Aufbau gibt. Da jedoch die Verbindung zum Server immer geöffnet bleibt, stellt dies keine besondere Beeinträchtigung für den Arbeitsprozess dar. Auch das Bestätigen und Neuanfordern verlorener und fehlerhafter Pakete kostet Zeit, da dies aber auf Protokollebene geschieht, ist dies sicherlich effizienter als es softwaretechnisch zu lösen.

Verschlüsselung und Sicherheit

Vor allem bei geschäftlichen Sitzungen oder Ideenfindungsprozessen ist auch die Vertraulichkeit der Verbindungen ein wichtiger Faktor. Im Prototypen wurde kein Verschlüsselungsverfahren eingebaut, dieses kann jedoch einfach auf den in .NET verwendeten `NetworkStream` aufgebaut werden. So kann dieser Datenstrom nach einmaligem Austausch von Verschlüsselungsparametern bzw. der öffentlichen Schlüssel durch einen `EncryptedStream` ausgetauscht werden, vgl. [3].

6.3.2 Whiteboard Server

Der Whiteboard Server ist die zentrale Steuer- und Verwaltungseinheit der Benutzer und Sessions. Das Whiteboard kann zwar auch ohne einer Internetverbindung benützt werden, jedoch kann dann keine der kollaborativen Funktionen genutzt werden. Beim Programmstart versucht die Anwendung automatisch sich beim zentralen Server anzumelden.

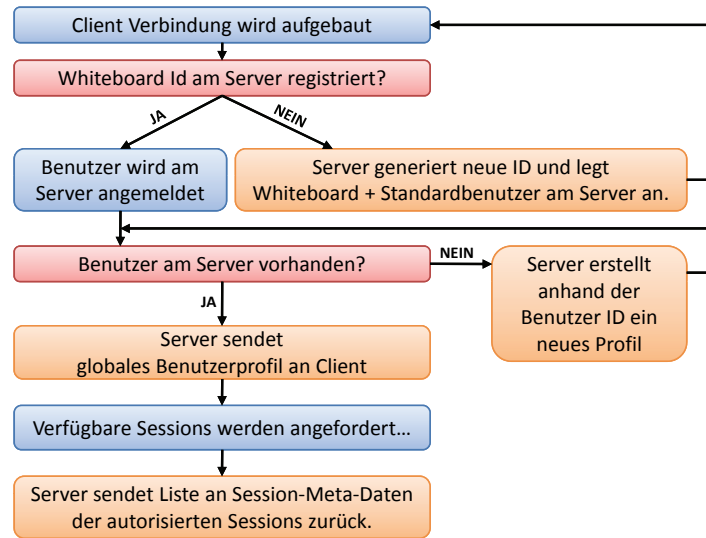


Abbildung 6.6: Ablaufdiagramm zur Anmeldung und Registrierung von Whiteboards und Benutzern.

Anmeldung und Authentifizierung

Wie schon in Abschnitt 4.2.2 beschrieben, sollte der Benutzer nichts vom Verbindungsaufbau und Loginprozess mitbekommen. Abbildung 6.6 zeigt wie der Authentifizierungsprozess vor sich geht.

Jedes Whiteboard, welches sich mit dem Server verbindet, benötigt eine eindeutige Whiteboard-*ID*, um mit anderen Whiteboards kommunizieren zu können. Besitzt ein Whiteboard noch keine eindeutige *ID*, so muss es sich bei der ersten Verbindung registrieren und der eindeutige Schlüssel wird in der Anwendungs-Konfigurationsdatei, welche pro Windows Benutzerkonto unterschiedlich ist, gespeichert. Um sich zu registrieren, muss dem Server lediglich ein eindeutiger Name mitgeteilt werden, sodann wird vom Server eine Whiteboard-*ID* generiert und dem Client zurückgesandt.

In ersten Tests wurde die aktive MAC-Adresse⁶ als Identifikationsschlüssel für das Whiteboard verwendet. Dies stellte sich allerdings als nicht geeignet heraus, da Geräte welche sowohl über einen LAN als auch einen WLAN Netzwerkadapter verfügten, beim Wechsel der Verbindung als zwei verschiedene Whiteboards erkannt wurden.

Mit Hilfe der pro Windows Benutzer eindeutigen Whiteboard-*ID* ist es nun möglich, Sessions einzelner Benutzer von denen anderer zu unterscheiden. Nicht zugelassene Benutzer haben keine Möglichkeit auf Sessions ande-

⁶Eine MAC-Adresse ist die Hardware-Adresse jedes einzelnen Netzwerkadapters. Diese wird zur eindeutigen Identifikation des Geräts im Netzwerk verwendet.

rer Benutzer zuzugreifen, solange diese nicht vom Besitzer der Session oder einem Administrator autorisiert werden.

Verwaltung von Sessions

Neben der Authentifizierung werden auf dem Server auch sämtliche Daten einer Session verwaltet. Alle Sessions, welche am Server angelegt werden, können lediglich von den authentifizierten Whiteboards eingesehen und beigetreten werden. Pro Session wird auch der gesamte Kommando-Stack gespeichert, welcher alle wichtigen Aktionen beinhaltet. Dieser kann so auf jedem Client auch im Nachhinein verwendet werden, um Aktionen rückgängig zu machen.

Sobald das erste Whiteboard eine Session betritt, wird diese aus dem Dateisystem in den Hauptspeicher des Servers geladen. Meldet sich das letzte Whiteboard von einer Session wieder ab, wird diese wiederum automatisch auf das Dateisystem gespeichert und aus dem Hauptspeicher gelöscht. Sobald eine Session läuft können sich alle authentifizierten Whiteboards mit dieser Session verbinden ohne weiteren Speicher zu belegen. Um einen Datenverlust vorzubeugen, wird die Session auch in einem bestimmten Zeitintervall gesichert.

Beim Verbinden mit einer Session fordern die Client Anwendungen den Kommando-Stack an (vgl. dazu Abschnitt 6.1.3). Dabei wird der gesamte Kommando-Stack mit den dazugehörigen Modellelementen übertragen. Somit werden auch die durch Ereignisse geänderten Daten berücksichtigt.

Die Netzwerkverbindung am Client

Die `ClientConnection` ist das Pendant zum Whiteboard Server in der Client Anwendung. Sobald die Anwendung gestartet wird, wird versucht eine Verbindung zum Server herzustellen. Ist dies erfolgreich geschehen, so kann sich die GUI oder der Controller des Clients auf die Ereignisse bzw. auf Nachrichten der Verbindung registrieren.

6.3.3 Nachrichtensystem

Um die Applikation bereits zur Entwicklungszeit so flexibel wie möglich zu gestalten wurde ein Nachrichtensystem entwickelt. Dies soll ermöglichen, dass die Netzwerkverbindung immer einheitliche Pakete sendet, welche aber dennoch flexibel genug sind um sie für verschiedenste Aufgaben und Nachrichten zu verwenden.

Eine Nachricht (`Message`) implementiert immer `IStreamable` und kann neben einem Nachrichtentyp auch noch weitere Objekte oder Binärdaten enthalten. Alle angehängten Objekte sollten ebenfalls `IStreamable` für einen schnellen Transfer implementieren. Ist dies nicht der Fall, so kann dieses Objekt nicht übertragen werden und wird übersprungen.

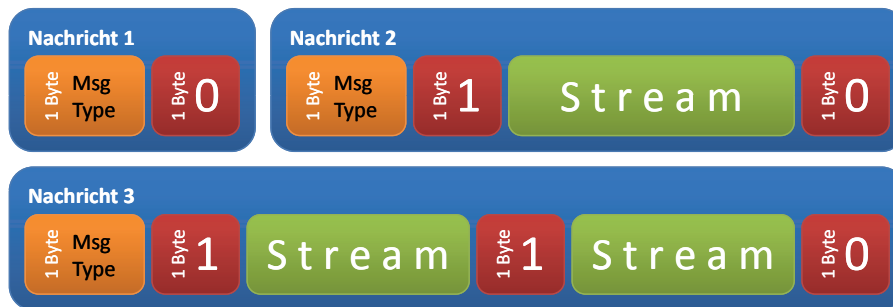


Abbildung 6.7: Nachrichten mit verschiedenen Inhalten.

Wie Abbildung 6.7 zeigt, besteht eine Nachricht im Idealfall aus nur zwei Bytes, wobei das erste Byte den Nachrichtentyp beschreibt. Eine Auflistung von möglichen Nachrichtentypen wird in Tabelle 6.1 dargestellt. Besitzt die Nachricht noch angehängte Objekte (z. B. Zeichenketten, Punkte, Kommandos, Ereignisse, ...) so wird im zweiten Byte 1 übertragen, was signalisiert, dass noch ein Datenpaket dieser Nachricht angehört. Wird 0 übertragen, so ist diese Nachricht komplett.

Serialisieren von Objekten

Der `StreamManager` ist eine Hilfsklasse zum schnellen Serialisieren bzw. Deserialisieren von Datenströmen. Alle zu serialisierenden Klassen müssen dazu beim `StreamManager` registriert werden. Nur so ist ein möglichst kleiner *Overhead* nötig, da die genaue Bezeichnung der Objekte, bzw. Klassen nicht übermittelt werden muss. Es wird lediglich eine zur Laufzeit bestimmte *ID*, welche nur ein Byte groß ist (und somit in dieser Ausbaustufe 256 verschiedene Klassen und Typen erlaubt), erzeugt. Dafür ist es notwendig, dass sowohl die Client-Anwendung, als auch der Server dieselbe Version des Netzwerk-*Assemblys* verwenden. Durch die Signierung des *Assemblys* und der Überprüfung dieser kann gewährleistet werden, dass nur Client-Anwendungen mit der richtigen DLL-Version sich am Server anmelden können.

Nachrichten, Antworten und deren Verteilung

Alle Nachrichten werden asynchron übertragen. Wird keine Antwort von dieser Nachricht erwartet (z. B. wenn lediglich ein Ereignis ausgesandt wird) so kann diese einfach mit Hilfe folgender Methode in die Warteschlange für Nachrichten eingereiht werden:

```
1 public void SendAsync(Message msg) {
2     messages.Enqueue(msg);
3 }
```

Tabelle 6.1: Auswahl einiger Nachrichtentypen.

<i>System</i>	<i>User/Session</i>	<i>In Session</i>
Idle	Register	WhiteboardConnected
Ok	Login	WhiteboardDisconnected
True	Logout	RequestCommandStack
False	CreateUser	ExecuteEvent
Successful	CreateSession	ExecuteCommand
Failed	RequestSessions	Undo
Latency	JoinSession	Redo
Undefined		LeaveSession

Zusätzlich dazu gibt es auch die Möglichkeit dem Nachrichtenobjekt einen Delegaten⁷ mitzugeben, welche nach dem Senden der Nachricht mit der Ergebnismeldung aufgerufen wird.

```

1 public void SendAndReceiveAsync(Message msg, AsyncCallback callback) {
2     msg.Callback = callback;
3     messages.Enqueue(msg);
4 }

```

Wie schon in den obigen Beispielen gezeigt, werden die Nachrichten in einer Warteschlange eingereiht, wobei Nachricht für Nachricht abgearbeitet wird. Sind keine Nachrichten mehr zu verarbeiten, dann wird die `Idle` Nachricht gesendet, sowie eine Latenz-Messung durchgeführt.

Wartet man nicht direkt auf die Antwort einer Nachricht, sondern möchte man generell alle Nachrichten eines Nachrichtentyps (siehe Tabelle 6.1) bekommen, so kann man sich auf diese registrieren. Man wird sodann über ein Ereignis benachrichtigt. Dabei gilt es zu beachten, dass sich auf diese Weise auch mehrere Stellen im Programm für eine Netzwerknachricht interessieren können.

Datenkonsistenz

Die Datenkonsistenz ist gewährleistet, in dem alle Kommandos welche auf einem Client erzeugt werden, zuerst am Server verifiziert werden bzw. das dortige Modell verändern. Erst danach werden sie wieder an alle Clients verteilt (auch an den der das Kommando erzeugt hat). Somit ist auch die Reihenfolge, in der die Kommandos abgearbeitet werden, sichergestellt, da sie von einem Punkt aus an die Clients verteilt werden.

⁷Ein Delegat ist ein typischer Zeiger auf eine Funktion. Dieser wird wie eine *Callback*-Methode verwendet, vgl. [10, Kapitel 22]

Wie bereits beschrieben, sollte der Benutzer allerdings keinerlei Verzögerung feststellen können, daher werden alle Aktionen beim aktiven Benutzer lediglich in der Präsentationsebene der Client-Anwendung verändert. Erst wenn das Kommando wieder vom Server zurück gesendet wird, wird das Datenmodell verändert und die Präsentationsebene aller anderen Clients aktualisiert bzw. die des aussendenden Client validiert.

6.3.4 Dokumentenaustausch

Um größere Dokumente bzw. Mediendaten auszutauschen wird neben dem Verwaltungs-Server ein Web- bzw. FTP-Server verwendet. Somit kann auf bereits verfügbare Server, welche für große Lasten ausgelegt sind, zurückgegriffen werden. Sofern es der Webserver unterstützt ist das *Streamen* von Videos ebenso möglich. Außerdem unterstützt WPF standardmäßig das Laden von Dateien (z. B. Bilder) von einem Webserver (siehe Abschnitt 6.4.2), was den Einsatz dieser Methode zusätzlich bestärkt.

Im Testaufbau werden dazu die *Microsoft Internet Information Services 7.0 (IIS)* eingesetzt. Die Implementierung im Client ist allerdings davon unabhängig.

6.4 Benutzeroberfläche

Durch die Umsetzung mit der *Windows Presentation Foundation* ist es möglich ein und dieselbe Applikation sehr einfach auf verschiedenen Plattformen bzw. Installationen zu verwenden. So kann die Whiteboard Prototypapplikation auf jedem PC laufen und mit Maus und Tastatur bedient werden. Natürlich kann anstatt des Monitors auch das generierte Bild auf eine Projektionsfläche geworfen werden und dort direkt mit digitalen Stiften bedient werden. Des Weiteren gibt es auch noch eine dritte Möglichkeit, und zwar den Einsatz der Software auf einen *TabletPC*⁸, welcher mit Hilfe eines *StylusPen* ebenfalls eine direkte Eingabe ermöglicht.

Wie bereits in Abschnitt 5.2.2 beschrieben, ist es in WPF möglich eigene Steuerelemente zu definieren. Um die Funktionalität von *Intoi* nachzubilden war es nötig einige Steuerelemente selbst zu entwerfen, um dasselbe Verhalten in WPF zu erzielen.

Der große Vorteil eigener Steuerelemente in WPF ist, dass diese sehr einfach im XAML Code eingefügt und so auch in anderen Applikationen verwendet werden können.

```
1 <WbCanvas Width="1024" Height="768" DoubleClick="OnDoubleClick" />
```

⁸Als *TabletPC* bezeichnet man einen tragbaren stiftbedienbaren Computer, der unter anderem wie ein Notizblock verwendet werden kann.

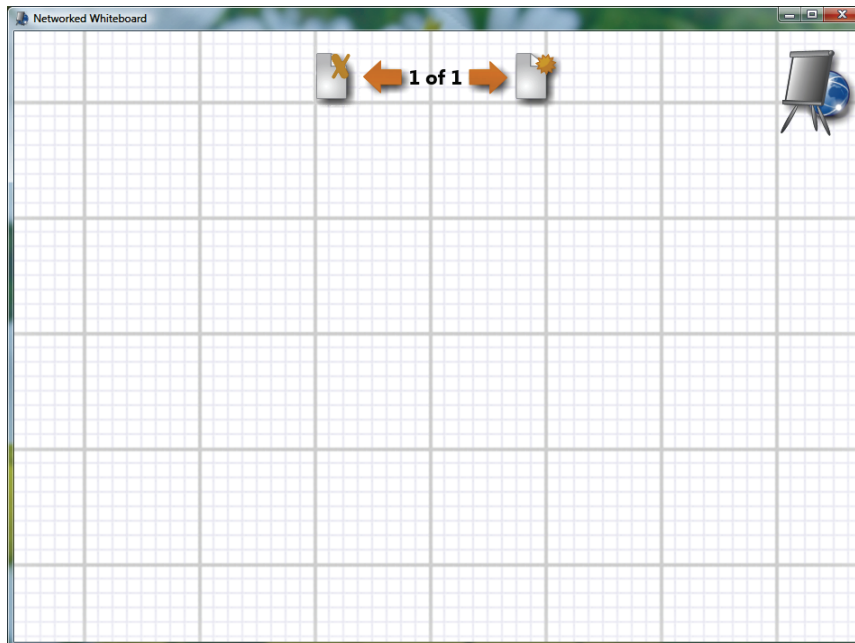


Abbildung 6.8: Nach dem Start der Applikation ist die Zeichenfläche sichtbar, welche beliebig in alle Richtungen verschoben, sowie vergrößert oder verkleinert, werden kann.

Kernstück der Applikation ist die im obigen Code-Beispiel eingebundene grenzenlose Zeichenfläche (`WbCanvas`), auf welche im nächsten Abschnitt eingegangen wird.

Alle eigenen Steuerelemente in der Whiteboard Anwendung beginnen mit dem Präfix `Wb`⁹, um sie von etwaigen gleich benannten Elementen aus dem .NET Framework einfach unterscheiden zu können.

6.4.1 Grenzenlose Arbeitsfläche

Die Idee zu einer grenzenlosen Zeichenfläche kam daher, dass dies eine der größten Einschränkungen bei einer Schultafel oder einem Flipchart ist. Wenn man am Ende der Zeile angekommen ist, gibt es keine Möglichkeit mehr etwa eine Grafik an einer der Randkanten weiterzuführen. Auch in herkömmlichen Computer-Anwendungen muss im Regelfall zuerst die Größe einer Grafik oder eines Dokuments bestimmt werden, und kann in der Regel nicht mehr (während des Arbeitsprozesses) verändert werden.

Daher wurde bereits in Intoi (vgl. Abschnitt 3.2.3) dieses Konzept umgesetzt und von den Benutzern als sehr gut und intuitiv empfunden, siehe dazu auch Kapitel 7. Abbildung 6.8 zeigt die grenzenlose Arbeitsfläche, wie

⁹`Wb`, Abkürzung für Whiteboard.

sie sich dem Benutzer nach dem Programmstart präsentiert.

Funktionalität

Das Steuerelement, welches die grenzenlose Arbeitsfläche implementieren soll, muss folgende Grundfunktionalitäten unterstützen:

- Container für verschiedene Elemente
 - Striche
 - Bilder, Grafiken
 - *PowerPoint*, *PDF* Dateien
- Sensitive Oberfläche für verschiedene Eingabemöglichkeiten
 - Digitaler Stift (vgl. Abschnitt 5.1.1)
 - *StylusPen* (*TabletPC*)
 - Maus
- Möglichkeit zur Navigation auf der Zeichenfläche
 - *Zoom & Pan*
 - Verschieben der Elemente

Neben der Sensitivität der Oberfläche und der Verarbeitung der von Eingabeereignissen verschiedener Eingabegeräten, müssen diese unter Umständen auch weitergegeben werden. So zum Beispiel, wenn ein Doppelklick auf der Oberfläche ausgeführt wird, muss dieses Ereignis an interessierte andere Programmteile, wie in diesem Fall der Menüebene, weitergegeben werden.

6.4.2 Elemente auf der Arbeitsfläche

Die folgenden Absätze beschreiben die Elemente welche auf die Zeichenfläche geladen werden können, sowie deren Anforderungen und Besonderheiten.

WbStroke - Kapselung eines Striches

Die WPF bietet zwar bereits fertige Steuerelemente (**InkCanvas**) an, um Strichdaten (**StrokeCollections**) zu rendern, allerdings ist es mit diesen nicht möglich Strichdaten mit anderen visuellen Elementen zu verbinden. Dies wäre lediglich insofern möglich, wenn Striche in eigenen Containern zwischen den eigentlichen Elementen abgelegt werden. Eine Verbindung mit anderen Multimedia Elementen wäre so nicht möglich.

Des Weiteren sind Strichdaten auch komplexer aufgebaut. Im Gegensatz zu einfachen Linien oder Polygonen, welche ebenfalls aus einer Abfolge



Abbildung 6.9: Bild-Dateien: Beim Upload (links) kann das Bild sofort angezeigt werden, beim Download (rechts) wird ein halbtransparenter Platzhalter angezeigt, solange das Bild übertragen wird.

von Punkten bestehen, kann ein Strich für jeden Punkt zusätzliche Eigenschaften, z. B. über den Anpressdruck, speichern. Die `DrawingAttributes` (Eigenschaften zum Zeichnen) bestimmen darüber hinaus die grundsätzliche Strichstärke, sowie dessen Farbe und die Behandlung der Enden (spitze oder abgerundete Enden). Alle diese Eigenschaften sind im WPF `Stroke` Element gekapselt.

Allerdings kann dieses `Stroke` Element nicht direkt (wie ein User Interface Element) gerendert werden, da es lediglich als Datenkapselung zum Speichern des Striches verwendet wird.

Die `WbStroke` Klasse übernimmt somit diese Zeichen-Funktionalität und bietet darüber hinaus noch Funktionen an um einen Strich hervorzuheben oder ihn als aktiv zu markieren. Ein Nachteil ist, dass solange ein Strich gezeichnet wird, er jedes mal komplett neu gezeichnet werden muss, was bei langen Strichen zu Leistungseinbußen führen kann.

WbFile/WbImage - Kapselung und Upload großer Daten

Wird auf der grenzenlos großen Zeichenfläche ein Bild oder eine andere große Datei geöffnet, so wird diese ebenfalls auf einen Server geladen (dazu siehe Abschnitt 6.3.4) und so zu den anderen Teilnehmern verteilt.

Um diesen Up- und Download kümmert sich zwar die dahinterliegende Modellschicht, das Öffnen eines Bildes und die Visualisierung des Fortschritts übernehmen aber die `WbFile` (allgemein für alle Dateien) bzw. die `WbImage` Klasse.

Der Uploader (in der Regel ein FTP-Client) sendet dazu die Fortschrittsmeldungen dem visuellen Anzeigeelement (siehe Abb. 6.9, links). Für den Datei-Download bietet sich eine weitere Funktion aus WPF an – dem automatisierten Dateidownload via HTTP. Somit kann das `WbImage` Objekt eine

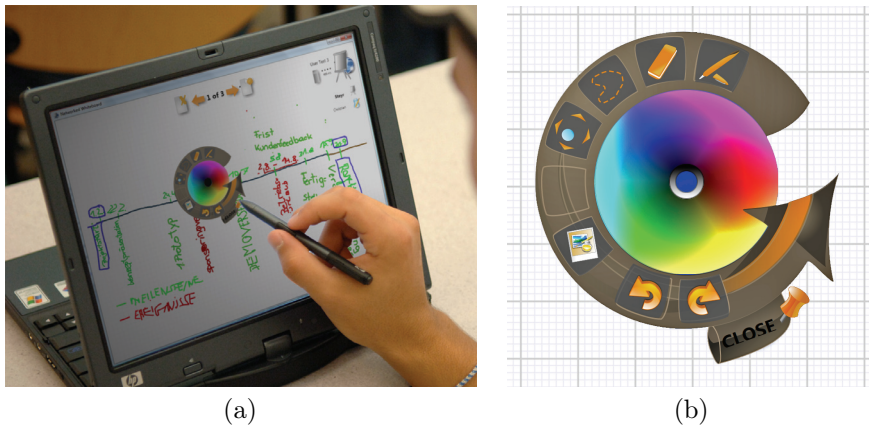


Abbildung 6.10: Werkzeugmenü: Das Menü kann (a) mit einem *Stylus-Pen* (direkte Eingabe) oder (b) mit einer Maus (indirekte Eingabe) bedient werden.

Verbindung zu einem Webserver herstellen, das Datenmodell kennt lediglich die eindeutige URL¹⁰ des Bildes, sowie dessen Größe und Position, und ob es bereits auf den Webserver übertragen wurde. Beim Download kann ebenfalls der Fortschritt abgefragt und visualisiert werden.

Kann das Bild (noch) nicht dargestellt werden, da es am Server noch nicht existiert, so wird ein Platzhalter angezeigt (siehe Abb. 6.9, rechts). Durch eine Benachrichtigung wird der Downloadprozess gestartet.

6.4.3 Menükonzept

Im Prototypen gibt es zwei Arten von Menüs, welche in diesem Abschnitt erklärt werden.

Werkzeugmenü

Das Werkzeugmenü wurde wiederum aus dem Projekt „*Intoi*“ übernommen, welches bereits in Abschnitt 3.2.3 betrachtet wurde. Der große Vorteil dieses Menüs ist, dass der Benutzer es auch auf großen Arbeitsflächen immer an der Position aufrufen kann wo es benötigt wird. Des Weiteren eignet sich das kreisrunde Menü besonders gut für die direkte Eingabe mit einem digitalen Stift. Abbildung 6.10(a) zeigt das Menü wie es auf eine *TabletPC* bedient werden kann, in (b) wird eine Großaufnahme des Menüs gezeigt.

Wie bereits erwähnt, wird das Menü durch einen Doppelklick aufgerufen. Diese Geste kann sowohl mit der Maus als auch mit dem Stift leicht ausgeführt werden. Das Menü wird rund um den Maus-Cursor oder dem

¹⁰ *Uniform Resource Locator*, beschreibt den Pfad zu einer Ressource über ein Netzwerkprotokoll.

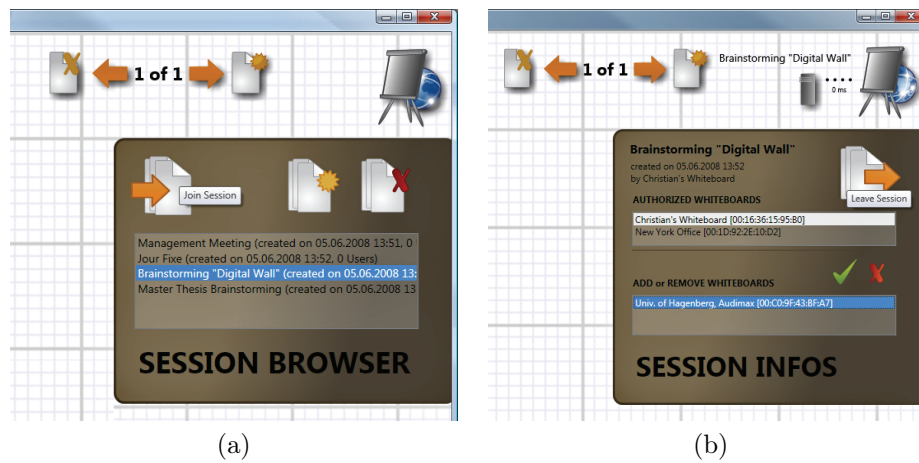


Abbildung 6.11: Session Management: Der *SessionBrowser* (a) zeigt alle autorisierten Sessions an. Eine Übersicht über die Session und der autorisierten Whiteboards liefert (b) das *SessionInfos* Panel.

Stift angezeigt. Dadurch sind die Distanzen zu den einzelnen Buttons bzw. Icons alle gleich weit vom Cursor weg. Wird das Menü auf einer Projektion verwendet, kann die Größe an die der einer durchschnittlichen Hand angepasst werden. Dadurch können die Buttons lediglich durch Bewegungen im Handgelenk erreicht werden.

Session Management

Im Gegensatz zum Werkzeugmenü gibt es auch noch einen zweiten Menübereich. Dieses Menü steht nur dann zur Verfügung, wenn die Whiteboard Applikation sich mit dem Server verbunden hat. Daraufhin wird das Menü automatisch aufgeklappt oder kann auch manuell durch Klick auf das Whiteboard in der rechten oberen Ecke aufgerufen werden.

In diesem Menü können die Sessions verwaltet werden, es ist möglich neue Sessions anzulegen bzw. zu löschen und einer Session beizutreten (siehe Abb. 6.11(a)). Sobald man einer Session beigetreten ist können im selben Bereich Einstellungen zur Session vorgenommen werden. So können z. B. weitere Whiteboards und Benutzer autorisiert werden. Dies ist in Abbildung 6.11(b) zu erkennen.

6.4.4 Eingabedialoge und Handschriftenerkennung

Es ist auch nötig Eingaben des Benutzers zu verarbeiten, wie z. B. den Namen des Benutzers oder der Session. Da bei dieser Anwendung bei der Stifteingabe auf Tastatur und Maus verzichtet werden soll, wurden spezielle Eingabedialoge geschaffen. Dieser, wie in Abbildung 6.12 dargestellt, besitzt ein

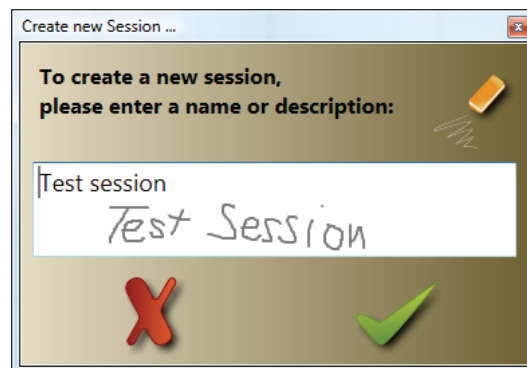


Abbildung 6.12: Dieser Eingabedialog kann sowohl mit Maus und Tastatur als auch mit einem Stift bedient werden.

etwas größeres Eingabefeld, welches in erster Linie wie ein normales Textfeld fungiert, das mit Tastatur und Maus bearbeitet werden kann. Des Weiteren liegt über diesem Textfeld auch noch eine transparente *InkCanvas*, mit der es möglich ist Stricheingaben vom digitalen Stift oder eines *StylusPen* zu verarbeiten. Dazu wird eine Gestenerkennung bzw. eine Handschriftenerkennung verwendet. Die erkannten Zeichen werden sofort als Text visualisiert.

Zur Erkennung wird das in *Windows Vista* bzw. *Windows XP Tablet-PC* bereits eingebaute Handschriftenerkennungssystem verwendet, vgl. [21]. Dieses System arbeitet mit *Time Delayed Neural Network*¹¹ Klassifikatoren, sowie einer „lernbaren“ Einheit welche der Benutzer trainieren kann. Standardmäßig stehen dabei bereits einige Wörterbücher in den gängigsten Sprachen, wie Englisch, Deutsch und Spanisch, aber auch Chinesisch und Japanisch zur Verfügung.

6.4.5 Kollaborative Funktionalitäten

Im letzten Abschnitt dieses Kapitels wird die Implementierung der Funktionalitäten erläutert, welche gemeinsames Arbeiten ermöglichen bzw. erleichtern sollen.

Wie bereits im Konzept (Abschnitt 4.2.4) beschrieben, muss es möglich sein, deutlich zu erkennen wenn ein anderer Benutzer Aktionen auf der Arbeitsfläche ausführt. Vor allem beim Schreiben kann es sehr leicht passieren, dass man entweder etwas neu Geschriebenes übersieht oder man sich beim Schreiben gegenseitig stört. So wird beim Zeichnen eines Striches dieser bei allen anderen verbundenen Whiteboards hervorgehoben. Beim Ausführen einer Aktion wird dies in der Benutzerliste rechts oben mit einem Werkzeug-Icon angezeigt (siehe Abb. 6.13, Benutzer Christian).

¹¹Neurale Netze kommen aus der künstlichen Intelligenz und sind Netze aus künstlichen Neuronen, die die Informationsverarbeitung abstrahieren sollen.

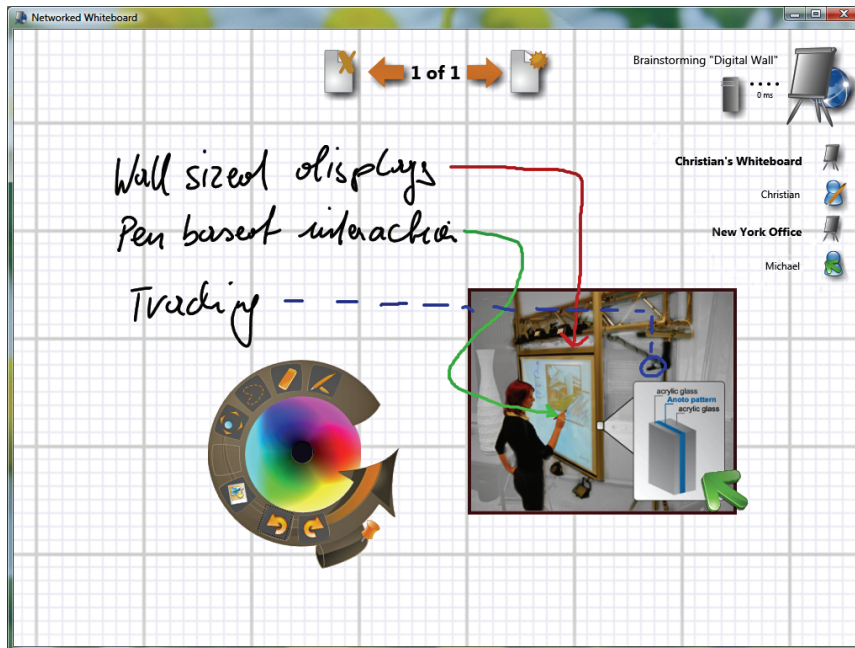


Abbildung 6.13: Ein Ausschnitt aus einem kollaborativen Arbeitsprozess am Whiteboard.

Wird mit Multimedia-Elementen interagiert, so muss ersichtlich sein sobald ein Benutzer ein Element ausgewählt hat. Dies geschieht durch eine einfache Änderung der Rahmenfarbe. Sobald ein Objekt ausgewählt worden ist, wird es für die anderen Benutzer gesperrt, und diese können das Objekt nicht mehr verschieben. Nach einem kurzem Zeitraum (z. B. 20 Sekunden, einstellbar) wird die Sperre automatisch aufgehoben. Beim Verschieben der Elemente wird lediglich der Zielpunkt der Verschiebung übertragen. Somit kann bei den anderen Clients die Verschiebung simuliert werden. Dies ist insofern wichtig, da man sonst den Überblick durch „herumspringende“ Bilder verlieren könnte, und man so genau weiß, woher welches Bild kommt.

Zu guter Letzt ist es möglich mit Hilfe eines Zeigewerkzeuges auf bestimmte Objekte oder Teile zu deuten und mit auswählbaren Icons andere Benutzer zur Interaktion oder zur Hilfestellung auffordern (siehe Abb. 6.13, Benutzer Michael).

Kapitel 7

Evaluierung und Diskussion

Um die Praxistauglichkeit, des in dieser Arbeit beschriebenen Prototypen, zu untersuchen, wurde ein Test mit verschiedenen Benutzern durchgeführt. Der Test selbst, sowie die Ergebnisse und Erkenntnisse werden in diesem Kapitel diskutiert.

7.1 Informeller Benutzertest

Bei dem durchgeführten Test handelt es sich um eine qualitative Benutzerstudie. Dabei haben Testpersonen Aufgaben am Prototypen bewerkstelligen müssen, wurden danach über ihre Erfahrungen und Probleme befragt und mussten einen Fragebogen beantworten. Auch die Beobachtungen, welche während dem Arbeiten gemacht wurden, fließen in diese Diskussion mit ein.

7.1.1 Testteilnehmer

Der Test wurde mit acht Teilnehmern zwischen 23 und 30 Jahren, sowie zwei Teilnehmern über 40 Jahren durchgeführt. Alle Testperson sind durchschnittliche bis erfahrene Computer-Nutzer gewesen, und vor allem mit *Office*-Anwendungen vertraut. Die Hälfte der Personen war weiblich. Lediglich zwei Personen hatten bereits mit einer digitalen Oberfläche und deren Bedienung Erfahrung gehabt.

Die Testpersonen wurden paarweise getestet, es arbeiteten immer zwei Personen an einem Szenario. Dabei verwendete eine Testperson das digitale Whiteboard, die Zweite bediente die Applikation wahlweise über Maus und Tastatur oder über ein Grafiktablett.

7.1.2 Testaufbau

Das Testsystem, welches mit Stiften bedient wurde, bestand aus einem *Acer Travelmate 8200* Notebook, welches einen 2 GHz *Core Duo* Prozessor, sowie 1 GB RAM und eine *ATI X1600* Grafikkarte besitzt. Als Betriebssystem

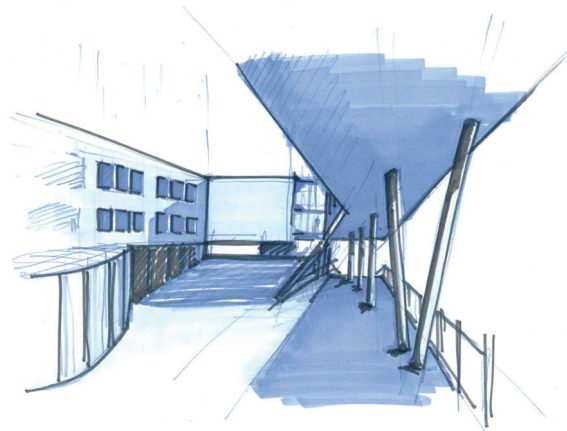


Abbildung 7.1: Diese Skizze diente als Inspiration für eine gemeinsame Arbeit am digitalen Whiteboard. (Grafik von Mag. Roland Keil)

wurde *Windows Vista* eingesetzt. Die *Anoto*-Stifte *DP-201* von *Maxell* sind mit dem integriertem *Bluetooth* Adapter verbunden. Die Installation selbst ist wie in Abschnitt 5.1 beschrieben aufgebaut. An der Installation wurde des Weiteren ein Mikrofon und zwei Lautsprecher angebracht, um eine Kommunikation mit der anderen Person zu ermöglichen. Das Notebook ist über eine drahtlose Internetverbindung mit dem Server verbunden.

Das zweite Testsystem bestand aus einem Desktop Computer (3 GHz *Pentium 4* Prozessor, 512 MB RAM) mit *Windows XP*, an den ein Grafiktablett sowie Maus und Tastatur angeschlossen wurden. Der Teilnehmer konnte über ein *Headset* mit der anderen Person kommunizieren. Die Audio-Verbindung wurde über *Skype* hergestellt.

Der Testserver wurde auf einem virtualisiertem *Windows Server 2008* installiert. Dieser Server, ein 2,9 GHz *AMD Athlon 64* Prozessor mit 512 MB RAM, befindet sich in einem Rechenzentrum in Nürnberg, Deutschland.

7.1.3 Testszenarios

Jedes Szenario sollte möglichst kurz gehalten werden und nicht länger als 15 Minuten dauern. Die Teilnehmer konnten aus folgenden Szenarien wählen. Manche Testpersonen führten auch beide Szenarien durch.

Kreative Aufgabe Dabei bekommen beide Testpersonen eine Vorlage einer einfachen handgefertigten Skizze eines Gebäudes (siehe Abb. 7.1). Diese gilt es in möglichst kurzer Zeit gemeinsam nachzuzeichnen. Des Weiteren sind dem Gebäude Beschriftungen hinzuzufügen. Bei dieser Aufgabe sollte vor allem die Problematik des geteilten Arbeitsbereiches untersucht werden.

Projektplanung Bei diesem Szenario gilt es eine Übersicht über einen Projektverlauf zu erstellen. Dabei bekommt jede Testperson eine Liste mit unterschiedlichen Meilensteinen, Ereignissen und Fristen, die auf einem Kalender oder einer Zeitleiste eingetragen werden sollten. Hierbei sollte es vor allem notwendig sein die grenzenlose Zeichenfläche zu verwenden.

7.1.4 Beobachtungen und Fragebogen

Die Testpersonen wurden während des Tests beobachtet und ihre Interaktionen mit dem digitalen Whiteboard wurden analysiert. Des Weiteren mussten sie nach dem Test am Prototypen folgende Fragen beantworten:

- Alter und Geschlecht?
- Würden Sie sich als erfahrenen Computernutzer einschätzen?
 - Ihre am häufigsten genutzten Anwendungen sind?

Kollaboratives distanziertes Arbeiten

- Haben Sie zumindest ein *Instant Messaging* Programm (z. B. *Skype*, *ICQ*, *MSN*) installiert? Haben Sie dieses auch schon einmal geschäftlich genutzt?
- Denken Sie, dass eine solche Software eine Besprechung oder ein gemeinsames Arbeiten am selben Ort ersetzen kann? Bitte begründen Sie Ihre Antwort.
 - Könnten Sie sich vorstellen eine ihrer nächsten Besprechungen über eine *VoIP* Anwendung (z. B. *Skype*) zu führen? Was würde Ihnen dabei fehlen? Was würde Sie unterstützen?

Interaktion mit dem Prototypen

- Haben Sie schon einmal mit einer digitalen Tafel oder einem digitalen Tisch interagiert, sei es mit Fingern oder Stiften?
- Wie ist es Ihnen mit der Stiftbedienung ergangen? Was hat Ihnen gefehlt?
- Wie haben Sie den Anmeldeprozess der Whiteboards bzw. die Auswahl der Sessions empfunden?
- Wie haben Sie die Interaktion mit den anderen Personen empfunden? Was war für Sie störend?
- Haben Sie immer den vollen Überblick über das aktuelle Geschehen am Whiteboard gehabt?
- Wie würden Sie die Applikation abschließend bewerten? Was hat Sie am meisten gestört? Was hat Ihnen am besten gefallen?

7.2 Ergebnisse und Erkenntnisse

7.2.1 Meinungen, Probleme und Ideen

Folgende Abschnitte fassen die durch den Autor beobachteten Testszenarien, sowie die Meinungen, Probleme und Ideen der Testteilnehmer zusammen.

Digitaler Stift im Vergleich zur Maus

Da die Anwendung für digitale Stifte ausgelegt ist, war es für die Teilnehmer, welche die Maus verwendet haben, natürlich um einiges schwieriger zu schreiben oder zu zeichnen. Es waren sich alle einig, dass die Maus dabei ungeeignet sei, und es eine Möglichkeit geben muss mit der Tastatur ebenfalls einen Text schreiben zu können.

Die Arbeit mit dem Stift wurde als „natürlich“, „abwechslungsreich“, „etwas Neues“ empfunden. Eine Testperson meinte, dass der digitale Stift viel besser in der Hand liegt als z. B. vergleichsweise eine Schulkreide oder ein Whiteboard-Marker. Die Latenz bei der Übertragung der Daten vom Stift zum Computer ist lediglich einer Testperson störend aufgefallen. Eine weitere Testperson hatte Probleme beim Schreiben an einzelnen Bereichen der Wand, da hier der Stift zu vibrieren begonnen hatte – dies signalisiert, dass das Punktmuster nicht gelesen werden kann (siehe Abschnitt 5.1.1).

Mehrere Testpersonen wünschten sich Buttons am Stift um ähnlich zur Maus einen Klick durchzuführen oder zwischen Werkzeugen zu wechseln. Es hatten mehrere Testpersonen ein Problem mit dem Doppelklick um das Menü aufzurufen.

Das Grafiktablett wurde nur von einer Testperson verwendet. Es eignete sich sehr gut zum Schreiben, kann aber eine Tastatur oder ein direktes Schreiben am Whiteboard nicht ersetzen.

Grenzenlose Fläche

Alle Testpersonen nutzten die grenzenlose Zeichenfläche intensiv. Es wurde allerdings als störend empfunden, wenn der Arbeitspartner, ohne es vorher anzukündigen, die Zeichenfläche verschoben hatte. Eine Idee wäre gewesen, dass die Zeichenfläche erst dann verschoben wird, wenn mehrere Sekunden lang keine Eingabe mehr geschehen ist.

Bei einem Testdurchlauf wussten die Testpersonen anfangs nicht, dass das Verschieben der Zeichenfläche auch die Fläche des Partners beeinflusst. Sie meinten es wäre sinnvoll, dies auswählbar zu machen, und man somit auch einen privaten Arbeitsbereich haben kann. Durch Klick auf das Whiteboard des Partners sollte ein einfaches Umschalten möglich gemacht werden.

Der Raster (siehe Abb 6.8) im Hintergrund diente den meisten Testpersonen als Orientierung und Hilfe. Bei einem Szenario wurde vor Arbeitsbeginn

das Hintergrundraster so verschoben, dass es als Monateinteilung genutzt werden konnte.

Das Menüsystem

Die Menüs wurden von den meisten als sehr gut bedienbar eingestuft, dabei gab es keine Unterschiede zwischen Maus- und Stiftbenutzer. Lediglich der Menüaufruf gelang mit dem Stift nicht immer, ein Knopf auf dem Stift sollte dieses Problem lösen.

Störend empfanden es einige Teilnehmer, nicht genau eine Farbe, bzw. dieselbe Farbe mehrmals zu verwenden. Auch wollten einige die Farbe des Arbeitspartners übernehmen, was nicht möglich war. Die meisten meinten, dass 16 oder 32 Farben für eine derartige Applikation anstelle des Farbrades (siehe Abb. 6.10) ausreichen würden.

Das Info-Panel (in dem die verbundenen Benutzer angezeigt werden) wurde von mehreren Testpersonen am Whiteboard nicht beachtet und als unnötig empfunden. Eine Idee war diese Benutzerinformationen ebenfalls verstreut auf der Arbeitsfläche anzuzeigen, und somit ausdrücken zu können wo man sich physisch aber auch gedanklich gerade befindet.

Die Schrifterkennung im Eingabedialog funktionierte nur dann ausreichend gut, wenn der Benutzer eine Schreibschrift verwendete. Druckbuchstaben wurden hingegen nur sehr schlecht erkannt, fast alle Testpersonen versuchten es aber zuerst mit Druckbuchstaben. Hierbei sollte die Schrifterkennung insofern verbessert werden, damit auch Druckbuchstaben ohne Probleme erkannt werden.

Rückgängig & Wiederholen

Die „Rückgängig“ Funktion wurde nur sehr selten verwendet, da das Radierer Werkzeug bevorzugt eingesetzt wurde. Vor allem bei vielen kurzen Strichen (so wie sie z. B. in geschriebenen Text vorkommen) dauert das „Rückgängig“ machen zu lange, da jeder Strich einzeln als „Rückgängig“ Kommando gespeichert ist. Eine Kapselung von Wörtern wäre hier wünschenswert. Des Weiteren waren sich die Testpersonen einig, lediglich seine eigenen Striche und Aktionen rückgängig machen zu wollen, und nicht auch die des Arbeitspartners.

Vermisste Funktionen

Die Testpersonen waren sehr anspruchsvoll und scheuten einen Vergleich mit professionellen Applikationen wie *Microsoft Word* oder *PowerPoint* nicht. Dadurch vermissten sie viele Funktionalitäten. Unter diesen waren z. B. umfangreichere Zeichenmöglichkeiten, wie Linien, Quadrate und Symbole. Auch war es schwierig an der Wand gerade Striche hinzubekommen. Des Weiteren

ren wollten zwei Testpersonen Bilder direkt aus dem Internet laden, was nur durch einen Wechsel der Applikation möglich war.

Das wahrscheinlich größte Manko, vor allem für den Benutzer am Desktop-Computer, war, dass keine Tastatureingaben möglich waren. Das die Handschrift nicht in eine Computerschrift umgewandelt wurde, wurde als gut empfunden, da es mehr an die Arbeit an einer Schultafel oder einem herkömmlichen Flipchart erinnerte. Bei einer Kombination aus Handschrift und Computerschrift standen die Tester eher skeptisch gegenüber, da „dies komisch aussehen könnte“.

Kollaboration

Die Sprachverbindung wurde von allen Testteilnehmern intensiv genutzt und als sehr wichtig empfunden. Dennoch wünschten sich die Meisten zusätzlich auch noch ein Videobild des Gegenübers zu sehen. Die Testpersonen waren sich einig, dass nur so Emotionen und Gedanken durch Gesichtsausdrücke etc. ausgedrückt werden können und das dies bei gemeinsamen Besprechungen sehr wichtig ist.

Interessant zu beobachten war ebenfalls, dass meist beide Partner simultan arbeiteten. Entgegen der Erwartung, dass die Testperson auf der Wand das Schreiben übernimmt, bemühten sich fast alle auch mit der Maus einigermassen gut ihren Text zu schreiben.

Die Benutzer an der Wand hatten Probleme den Überblick über das gesamte Geschehen zu behalten. Hingegen die Computernutzer orientieren sich am Info-Panel und konnten einfach dem Zeichnen des Partners folgen. Lediglich das Löschen von Strichen wurde als zu schnell und störend empfunden, da diese gelöschten Striche auf einmal weg waren. Einige meinten das solche Abläufe noch besser visualisiert und animiert werden sollten (z. B. durch Verschieben in einen Papierkorb).

Auch dauerte es manchmal sehr lange bis einzelne Striche anderer Benutzer auf der Fläche erschienen sind. Die Zeit in der die Striche des Arbeitspartners zuerst auf den Server übertragen, verteilt und dann in der Applikation angezeigt wurden, stellte sich als störend heraus.

Abschließende Meinungen

Fast alle Testpersonen sprachen sich von einer interessanten und neuen Erfahrung, mit der digitalen Wand zu arbeiten. Vor allem, dass man mit einem Computer interagiert, ohne davor zu sitzen wurde als sehr gut empfunden. Besonders bei kreativen Meetings ist es wichtig vom Tisch oder Arbeitsplatz wegzukommen und sich frei vor einer Wand bewegen zu können. Lediglich eine Testperson meinte, dass die Arbeit an der digitalen Wand anstrengender sei als vor dem Computer.

Obwohl eine Audioverbindung bestand, war dies aber den Testpersonen

Tabelle 7.1: Richtgrößen gängiger Datenpakete. Die tatsächliche Größe kann sich durch optionale Eigenschaften ändern.

<i>Nachricht/Element</i>	Gesamtgröße der Nachricht
JoinSession	16 Bytes
AddElementCommand	mind. 76 Bytes
MoveElementCommand	76 Bytes
RemoveElementCommand	42 Bytes
StrokePointEventArgs	36 Bytes
StrokeFinishedEventArgs	18 Bytes
StrokeModel (mit 100 Punkten)	2018 Bytes

nicht ausreichend um sinnvoll kommunizieren zu können. Die generelle Meinung war, dass die Persönlichkeit der einzelnen Besprechungsteilnehmer verloren geht, da kein unbewusstes bzw. non-verbales Feedback durch Gesten, Gesichtsausdrücke, etc. möglich sei.

Dennoch ist diese Anwendung durchaus dazu geeignet Besprechungen an verteilten Orten durchzuführen. Vor allem ist das digitale Whiteboard eine gute Ergänzung für Besprechungen und kreative Arbeitsprozesse. Eine ernsthafte Diskussion sollte aber immer noch von Person zu Person geführt werden. Dennoch ist es sehr praktisch wenn Besprechungsergebnisse später auf das eigene Whiteboard übertragen und weiterbearbeitet werden können. Die Applikation sollte allerdings noch weiter ausgebaut werden, um mit mehr Funktionen einen schon aus anderen Applikationen bekannten Komfort zur Verfügung zu stellen. Auch an den Latenzproblemen muss gearbeitet werden.

7.2.2 Leistungsevaluierung

Neben den Erfahrungen der Benutzer wurden auch die übertragenen Datenpakete bzw. Nachrichten analysiert. Bei der Implementierung wurde vor allem auf geringe Größen der Nachrichtenpakete geachtet. Tabelle 7.1 zeigt die durchschnittlichen Nachrichtengrößen bei einigen ausgewählten Kommandos und Ereignissen.

Auch längere Striche sollten somit kein Problem darstellen, da die einzelnen Punkte alle einzeln und kontinuierlich übertragen werden. Allerdings kommen die Punkte nicht kontinuierlich beim Endbenutzer an, was den Eindruck erwecken lässt, dass die Punkte nur stockend übertragen werden. Dies lässt sich aber auf das Buffern der Nachrichten zurückführen. Da dies störend wirkt, sollte – ähnlich wie bei *Multi-Player* Spielen – eine Latenzkompensation geschehen, vgl. [1].

Zum Analysieren des Netzwerk-Datenstromes wurde *WireShark*¹ verwendet. Eine Nachricht, welche zum Server hin und wieder zurück gesendet wird, benötigt demnach im Testaufbau zwischen 300 und 500 Millisekunden. Während einer 15-minütigen Sitzung wurde ein Datentransfer von ca. 1.300 KB von und zum Server verzeichnet. Dies entspricht einer durchschnittlichen Datenleistung von ca. 1.500 Bytes pro Sekunde ($\approx 0,012$ MBit pro Sekunde).

Die größten Pakete werden beim Laden einer bereits vorhandenen Session übertragen, da das gesamte Modell, sowie der Kommando-Stack übertragen wird. Beim Öffnen einer Session muss also fast der gesamte Datentransfer, welcher während der eigentlichen Sitzung angefallen ist, übertragen werden (bei obigen Beispiel ca. 1 MB). Dies könnte ebenfalls zusätzlich optimiert, bzw. der Datenstrom komprimiert werden.

¹*WireShark*, Programm zur Analyse von Netzwerk-Kommunikationsverbindungen, www.wireshark.org

Kapitel 8

Zusammenfassung

In dieser Arbeit wurde die Entwicklung einer vernetzten Whiteboard Anwendung diskutiert, mit Hilfe deren verteilte Arbeitsprozesse ermöglicht bzw. auch erleichtert werden sollen. Es gibt viele Visionen, die neue Arbeitsweisen für die Zukunft präsentieren. Vor allem die Globalisierung und der gesellschaftliche Wandel tragen dazu bei, dass es immer wichtiger wird schnellen und direkten Informationsaustausch zu betreiben, ohne dabei am selben Ort des Geschehens zu sein. So wie sich Konzerne über Kontinente verteilen, werden auch Arbeitsplätze verteilt sein.

Der aktuelle Stand der Technik zeigt, dass es bereits Technologien gibt, die ein kollaboratives Arbeiten ermöglichen, dennoch ist der Benutzer meist an einen Computer gebunden, welcher zusätzliche Hürden schafft. Die Idee mit Hilfe digitaler Whiteboards einen neuen verbesserten Arbeitsprozess zu kreieren wurde in dieser Arbeit ebenso vorgestellt. Untersuchungen realer Arbeitsprozesse zeigen wie eine Software in Teilbereichen diesen Ablauf unterstützen kann.

Der in dieser Masterarbeit vorgestellte Prototyp soll zeigen, dass gemeinsame Arbeit mit einer Whiteboard Anwendung möglich ist und zeigt dabei auch die Grenzen der Interoperabilität zwischen wirklichen digitalen Whiteboards und Desktop-Applikationen auf. Doch genau diese Möglichkeit Anwendungen auf unterschiedlichen Plattformen an jedem Ort mit möglichst geringen Systemanforderungen betreiben zu können, ist für den Erfolg einer solchen Arbeitslösung ebenso notwendig wie die gute Bedienbarkeit der Applikation.

Die Evaluierung des Protoypens zeigte, dass die Testpersonen, obwohl keiner schon einmal mit einem digitalen Whiteboard gearbeitet hat, dieses neue Arbeitsmodell sehr schnell angenommen haben. Dies ist vor allem der einfachen Bedienbarkeit der Anwendung zu verdanken. Die Testpersonen waren sich Testpersonen einig, dass eine solche Applikation aktuelle Anwendungsszenarien sehr gut unterstützen kann, für komplette Sitzungen müssen aber vor allem noch kommunikativere Funktionen eingebaut werden.

8.1 Ausblick und Wünsche

Natürlich ist es selbstverständlich, dass diese Implementierung des Prototypen nicht die derzeitigen Arbeitsweisen ändern kann, da sich digitale Whiteboards noch nicht am Markt durchgesetzt haben und es auch andere technische Hürden, wie z. B. die stetige Verfügbarkeit einer breitbandigen Internet-Verbindung, zu überwinden gibt. Dennoch ist klar erkennbar, dass die Technik sehr schnell voranschreitet. Digitale Whiteboards werden immer mehr in Besprechungsräumen, Klassenzimmer und auch in Fernsehstudios eingesetzt um Inhalte auf eine neue und moderne Art zu präsentieren.

Ein Problem ist sicherlich, dass es derzeit keine einheitlichen Standards gibt – die Stift- oder Finger-Tracking-Technologien sind allesamt unterschiedlich – und jede bringt seine eigenen Probleme mit sich. Vor allem die Genauigkeit und Schnelligkeit ist entscheidend um ein effizientes Arbeiten zu ermöglichen. Doch Betriebssysteme der nächsten Generation sollen *Multi-Touch*-fähige Eingabegeräte bereits auf Systemebene unterstützen.

Derzeit sind digitale Tische oder Whiteboards noch zu teuer, um sich am Massenmarkt durchzusetzen. Videoverbindungen über das Internet leiden immer noch an schlechter Qualität, da meist nur sehr kleine Bildgrößen und nur wenige Bilder pro Sekunde übertragen werden, um Echtzeitübertragungen zu ermöglichen. Die Bandbreiten werden laufend erhöht und auch mobile Breitbandinternetzugänge werden immer häufiger und beliebter.

8.2 Resümee

Wenn man nun zurück auf das Kapitel 2 – Visionen – blickt, und diese mit dem Prototypen vergleicht, kann man sehr schön erkennen, dass es zwar noch an vielen Funktionalitäten fehlt, generell aber die Grundideen umgesetzt wurden. Mit Hilfe dieser Anwendung ist es nun einfach möglich Verbindungen zwischen Geschäfts- oder Arbeitspartnern herzustellen, über die man mehr als nur Sprache austauschen kann. Direkte und gemeinsame Interaktionen auf einem digitalen Whiteboard wurden demonstriert. Der erfolgreiche Einsatz neuer Softwaretechnologien wurde ebenso gezeigt, wie die Umsetzung einer flexiblen Server/Client-Lösung, welche sich zum Ausprobieren neuer verteilter Konzepte und Anwendungen eignet.

Wie die gesamte Masterarbeit zeigt, steckt in dieser Thematik enorm viel Potenzial, da aufgrund der Kombination verschiedenster Technologien, sowie der Umsetzung neuartiger Bedienkonzepte und Menüoberflächen neue Software geschaffen werden kann. Dadurch können bereits bekannte Problematiken neu aufgerollt und auf eine neue Art und Weise umgesetzt werden.

Anhang A

Inhalt der CD-ROM

File System: Joliet

Mode: Single-Session (CD-ROM)

A.1 Masterarbeit

Pfad: /thesis

Masterarbeit_ChristianSchafleitner.pdf Masterarbeit (PDF)
/images Abbildungen
/videos In der Arbeit referenzierte Videos
/references Internet-Referenzen als PDF

A.2 Prototypische Anwendung

Ausführbare Programmdateien

Die folgenden ausführbaren Dateien wurden auf einem unter *Windows Vista* laufenden *Visual Studio 2008* mit dem *.NET Framework 3.5* kompiliert.

Pfad: /binaries

Whiteboard.exe Desktop/Client-Anwendung
Whiteboard.config Konfigurationsdatei
Whiteboard.Server.exe Server-Anwendung (Konsole)
Whiteboard.Service.exe Server-Anwendung (Windows Service)
*.dll Dazugehörige Programmbibliotheken

Quellcode

Der gesamte C# Quellcode inklusiver Grafiken und *Visual Studio 2008* Projektdateien befindet sich in folgenden Ordnern auf der CD-ROM.

Pfad: /source

/Whiteboard	Desktop/Client Anwendung
/Whiteboard.AnotoPen . . .	AnotoPen Bibliothek
/Whiteboard.AnotoUtilities .	Hilfsbibliothek um Stiftdaten zu lesen
/Whiteboard.Core	Kernprogrammbibliothek
/Whiteboard.Network	Netzwerkbibliothek
/Whiteboard.Server.Console .	Konsolen Server-Anwendung
/Whiteboard.Server.Service .	Windows Service
Whiteboard.sln	Visual Studio 2008 Solution File

Literaturverzeichnis

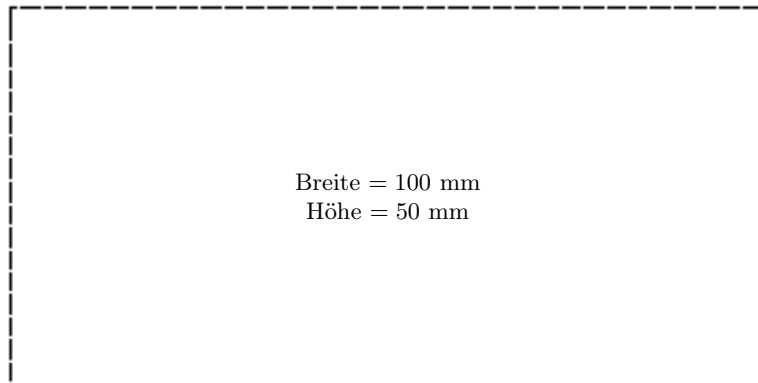
- [1] Armitage, Grenville, Mark Claypool und Philip Branch: *Networking and Online Games: Understanding and Engineering Multiplayer Internet Games*. John Wiley & Sons, Ltd., 2006, ISBN 0-4700-1857-7.
- [2] Baset, Salman A. und Henning G. Schulzrinne: *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol*. White Paper, Department of Computer Science, Columbia University, New York, September 2004.
- [3] Bless, Roland, Erik Oliver Blaß, Michael Conrad, Hans Joachim Hof, Kendy Kutzner, Stefan Mink und Marcus Schöller: *Sichere Netzwerkkommunikation: Grundlagen, Protokolle und Architekturen*. Springer, Berlin, 2005.
- [4] Brandl, Peter, Michael Haller, Michael Hurnaus, Verena Lugmayr, Juergen Oberngruber, Claudia Oster, Christian Schafleitner und Mark Billingham: *An adaptable rear-projection screen using digital pens and hand gestures*. In: *ICAT '07: Proceedings of the 17th International Conference on Artificial Reality and Telexistence*, Seiten 49–54, Washington, DC, USA, 2007. IEEE Computer Society, ISBN 0-7695-3056-7.
- [5] Callahan, J., D. Hopkins, M. Weiser und B. Shneiderman: *An empirical comparison of pie vs. linear menus*. In: *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 95–100, New York, NY, USA, 1988. ACM, ISBN 0-201-14237-6.
- [6] Forlines, Clifton, Daniel Wigdor, Chia Shen und Ravin Balakrishnan: *Direct-touch vs. mouse input for tabletop displays*. In: *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 647–656, New York, NY, USA, 2007. ACM, ISBN 978-1-59593-593-9.
- [7] Friedland, Gerald, Lars Knipping und Raúl Rojas: *E-Chalk: Technical Description*. Technischer Bericht, Freie Universität Berlin, Institut für Informatik, August 2001.

- [8] Gamma, Erich, Richard Helm, Ralph Johnson und John Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994, ISBN 0-201-63361-2.
- [9] Gownder, J.P. und James L. McQuivey: *The Future Of Apple Inc. - By 2013, Apple's Product Mix Will Make It A Credible Hub Of The Digital Home*. Forrester Research, 2008.
- [10] Gunnerson, Eric: *A Programmer's Introduction to C#*. Apress, 2000.
- [11] Hein, Mathias: *TCP/IP*. MITP-Verlag, 1998.
- [12] Microsoft: *Real-time Collaboration Future Vision*. Video, <http://www.youtube.com/watch?v=RiTb3-t4YJk> (auf CD-ROM), 2005.
- [13] Microsoft: *With Windows MultiPoint, Youths in Developing-World Classrooms Learn 21st-Century Skills*. <http://www.microsoft.com/presspass/features/2006/dec06/12-14MultiPoint.msp> (auf CD-ROM), Dezember 2006.
- [14] Microsoft: *Microsoft Launches New Product Category: Surface Computing Comes to Life in Restaurants, Hotels, Retail Locations and Casino Resorts*. <http://www.microsoft.com/presspass/press/2007/may07/05-29MSSurfacePR.msp> (auf CD-ROM), Mai 2007.
- [15] Microsoft, Industry Innovations Group: *Health Future Vision*. Video, <http://video.msn.com/video.aspx?vid=4f788377-2fd6-40e7-a9f3-5dd7959ef683> (auf CD-ROM), 2007.
- [16] Microsoft, Industry Innovations Group: *Manufacturing Future Vision*. Video, <http://video.msn.com/video.aspx?vid=2750da8a-21da-40f5-a171-0d3258f42683> (auf CD-ROM), 2007.
- [17] *Minority Report*. Regie: Steven Spielberg. Drehbuch: Philipp K. Dick, Scott Frank and Gary Oldman. Mit Tom Cruise, Colin Farrel, Max von Sydow, Steve Harris, Neal McDonough. DVD, 20th Century Fox/DreamWorks (2002).
- [18] Nathan, Adam: *Windows Presentation Foundation Unleashed*. Sams Publishing, 2007.
- [19] Okada, Ken Ichi, Fumihiko Maeda, Yusuke Ichikawaa und Yutaka Matsushita: *Multiparty videoconferencing at virtual social distance: MAJIC design*. In: *CSCW '94: Proceedings of the 1994 ACM conference on computer supported cooperative work*, Seiten 385–393, New York, NY, USA, 1994. ACM, ISBN 0-89791-689-1.

- [20] Pawar, Udai Singh, Joyojeet Pal, Rahul Gupta und Kentaro Toyama: *Multiple mice for retention tasks in disadvantaged schools*. In: *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 1581–1590, New York, NY, USA, 2007. ACM, ISBN 978-1-59593-593-9.
- [21] Pittman, James A.: *Handwriting Recognition: Tablet PC Text Input*. Computer, 40(9):49–54, 2007, ISSN 0018-9162.
- [22] Rekimoto, Jun: *SmartSkin: an infrastructure for freehand manipulation on interactive surfaces*. In: *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 113–120, New York, NY, USA, 2002. ACM, ISBN 1-58113-453-3.
- [23] Reyes, August de los: *A Design Vision for Microsoft Surface: Predicting the Past*. Vortrag, Mix Essentials, Siemensforum Wien, 19. Mai 2008.
- [24] Smith, Grahm, M. C. Schraefel und Patrick Baudisch: *Curve dial: eyes-free parameter entry for GUIs*. In: *CHI '05 extended abstracts on Human factors in computing systems*, Seiten 1146–1147, New York, NY, USA, 2005. ACM, ISBN 1-59593-002-7.
- [25] Stevens, W. Richard: *UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI*. Prentice Hall, 1998.
- [26] Tang, John C. und Scott Minneman: *VideoWhiteboard: video shadows to support remote collaboration*. In: *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 315–322, New York, NY, USA, 1991. ACM, ISBN 0-89791-383-3.
- [27] Tapia, Ernesto, und Raúl Rojas: *Recognition of On-line Handwritten Mathematical Expressions Using a Minimum Spanning Tree Construction and Symbol Dominance*. Springer Berlin / Heidelberg, 2004.
- [28] Watt, A., C. Lilley, D. Ayers, R. George, C. Wenz, T. Hauser, K. Lindsey und N. Gustavsson: *SVG Unleashed*. Sams Publishing, 2002, ISBN 0-6723-2429-6.
- [29] Wilson, Andrew D.: *TouchLight: an imaging touch screen and display for gesture-based interaction*. In: *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, Seiten 69–76, New York, NY, USA, 2004. ACM, ISBN 1-58113-995-0.

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —