



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Dennis Hollatz

Einsetzbarkeit von AJAX-basierten Applikationen
für kooperatives Arbeiten

Dennis Hollatz

Einsetzbarkeit von AJAX-basierten Applikationen
für kooperatives Arbeiten

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Jörg Raasch

Abgegeben am 27. Februar 2007

Dennis Hollatz

Thema der Bachelorarbeit

Einsetzbarkeit von AJAX-basierten Applikationen für kooperierendes Arbeiten

Stichworte

AJAX, Collaborative Workspace, Groupware, kooperierendes Arbeiten, leichtgewichtige Clients, Mensch-Computer-Interaktion, Pairware, Tabletop, Ubiquitous Computing, WEB 2.0

Zusammenfassung

Die Entwicklung auf dem Gebiet des computergestützten kooperierenden Arbeitens hat in den vergangenen Jahren große Fortschritte gemacht. Es gibt viele Projekte und Studien zu diesem Thema, welche sich jedoch meist auf einen einzelnen integrierten Raum (Collaborative Workspace) beziehen. Diese Arbeit zeigt eine Alternative zu so einem vollständig ausgestatteten Raum auf. Es wird eine Komponente vorgestellt, die sich in quantitativ größerem Umfang und dezentral in einem Unternehmen platzieren lässt. Zur Minimierung der Wartungskosten wird der Einsatz einer AJAX-basierten Webanwendung vorgestellt, welche es ermöglicht, Updates und Fehlerbeseitigungen direkt vom zentralen Server aus zu tätigen. Die AJAX-Technologie ermöglicht Webanwendungen ein ähnliches Verhalten, wie es bereits von klassischen Desktopanwendungen bekannt ist. Weiterhin wird in dieser Arbeit untersucht, ob sich die AJAX-Technologie ebenfalls für den Einsatz im Umfeld des kooperierenden Arbeitens mehrerer Personen am gleichen Ort eignet.

Dennis Hollatz

Title of the paper

Applicability of AJAX-based Solutions for Cooperative Work

Keywords

AJAX, Collaborative Workspace, Groupware, cooperative work, Thin-Clients, Human-Computer-Interaction, Pairware, Tabletop, Ubiquitous Computing, WEB 2.0

Abstract

The developments in computer supported collaborative work and collaborative workspaces made great progress during the past years. There are many studies and projects regarding this topic, they mostly aim for a centralised, integrated one-room environment, where you have the possibility to deploy a variety of rich clients. This thesis develops an alternative to such a complete, integrated and centralized approach, by presenting the concept for a component, which can be easily placed anywhere in an enterprise environment. To minimize maintenance costs this component will run AJAX based applications, which run in a Gecko based web browser, so they can be maintained and updated on a central web server. The usage of AJAX technologies makes it possible to develop web applications which have a similar behaviour to classical desktop applications. This thesis is addressed the the question if AJAX is suitable for the application in a face-to-face collaborative work environment.

*Alles, was ich denke, haben schon andere gedacht.
Alles, was ich schreibe, haben schon andere geschrieben.
Doch alles ist neu - durch die Verbindung*

*Karl-Josef Durwen
(Durwen, 2001)*

Inhaltsverzeichnis

Tabellenverzeichnis	9
Abbildungsverzeichnis	10
1 Einführung	11
1.1 Kontext und Motivation	11
1.2 Zielsetzung	13
1.3 Gliederung	14
2 Infrastrukturen für Collaborative Workspaces	15
2.1 Roomware	15
2.1.1 DynaWall	16
2.1.2 CommChair	16
2.1.3 InteracTable	17
2.1.4 ConnecTable	18
2.1.5 Zusammenfassung	18
2.2 iRoom	20
2.3 Der Collaborative Workspace an der Hochschule für Angewandte Wissen- schaften Hamburg	22
2.4 Zusammenfassung	22
3 Einsatzkontexte eines Collaborative Workspace	23
3.1 Digitales Whiteboard	23
3.1.1 Beispiel: BlueBoard (IBM)	25
3.2 Digitales Whiteboard in Kombination mit privaten Clients	26
3.2.1 Beispiel: Moderiertes Brainstorming	26
3.2.2 Beispiel: Einsatzleitstand	27
3.2.3 Bedeutung für die Architektur	28
3.3 Tabletop	28
3.3.1 Besonderheiten der Oberfläche	29
3.3.2 Beispiel: Physikbasierte Interaktion	31
3.3.3 Beispiel: Interface Currents (Uta Hinrichs)	31
3.3.4 Bedeutung für die Architektur	33

3.4	Zusammenfassung	33
3.5	Bewertung der vorgestellten Ansätze	34
4	AJAX-Konzepte	35
4.1	Hintergründe	35
4.2	Technologien	37
4.3	Beispiele	38
4.3.1	Google Maps	38
4.3.2	MindMapping (Virginio Carfagno)	38
4.3.3	AJAX-basierte Office Anwendungen	39
4.3.4	Meebo	40
4.4	Zusammenfassung und Fazit	40
5	Beispielszenario: Virtueller Leuchttisch zur Erstellung einer Fotozeitschrift	42
5.1	Randbedingungen	42
5.1.1	Qualifikation der Benutzer	43
5.1.2	Verteilung der Anwendung	43
5.2	Funktionale Anforderungen	43
5.2.1	Bereitstellen von Bild- und Textdaten	43
5.2.2	Zugriff auf hinterlegte Bild- und Textdaten	43
5.2.3	Einfügen von Bildern und Texten in den Layout-Bereich	44
5.2.4	Import und Export von Layoutdaten	44
5.2.5	Ändern der Orientierung des Layout-Bereichs	44
5.2.6	Transaktionsverwaltung	45
5.3	Nicht-funktionale Anforderungen	45
5.3.1	Förderung der Kommunikation	45
5.3.2	Anforderungen an Eingabegeräte	46
5.3.3	Anforderungen an die Usability	46
6	Systementwurf	47
6.1	Identifikation von Anwendungsfällen	47
6.2	Fachliche Architektur	49
6.2.1	Aufteilung in Teilsysteme	49
6.2.2	Oberflächenkonzept	51
6.3	Technische Architektur	54
6.3.1	Rahmenbedingungen	54
6.3.2	Einteilung in Schichten	55
6.3.3	Kommunikationsprotokoll	55
6.3.4	Anwendungsdesign	57
6.4	Realisierung eines Prototypen	58
6.4.1	Zielsetzung	58

6.4.2 Herausforderungen	59
6.4.3 Implementation	60
7 Beurteilung und Fazit	62
7.1 Zusammenfassung	62
7.2 Ausblick	63
Literaturverzeichnis	64
A Anwendungsfalltabellen	68
Glossar	78
Index	81

Tabellenverzeichnis

6.1 Anwendungsfall: Wechsel der Initiative	48
A.1 Anwendungsfall: Abbruch einer Transaktion	68
A.2 Anwendungsfall: Einfügen von Elementen in den gemeinsamen Bereich	69
A.3 Anwendungsfall: Verschieben eines Elements im gemeinsamen Bereich	70
A.4 Anwendungsfall: Entfernen von Elementen aus dem gemeinsamen Bereich	71
A.5 Anwendungsfall: Bereitstellen von Daten für den privaten Bereich	72
A.6 Anwendungsfall: Exportieren des erstellten Layouts	72
A.7 Anwendungsfall: Importieren von Layoutbeschreibungen	73
A.8 Anwendungsfall: Austausch von Elementen	74
A.9 Anwendungsfall: Anlegen einer neuen Zeitschriftenseite	75
A.10 Anwendungsfall: Entfernen einer Zeitschriftenseite	76
A.11 Anwendungsfall: Verschieben einer Zeitschriftenseite	77

Abbildungsverzeichnis

1.1	Beispiel: InteracTable	13
2.1	Übersicht der Roomware-Komponenten	16
2.2	Zwei Personen an einer DynaWall	17
2.3	Die zweite Generation des <i>CommChair</i>	18
2.4	Fraunhofer InteracTable	19
2.5	Zwei miteinander verbundene ConnectTables	19
2.6	Der <i>iRoom</i> in Benutzung	20
2.7	Schematische Darstellung des <i>iRoom</i>	21
3.1	Schematische Darstellung der Whiteboard-Metapher	24
3.2	Das Blueboard in Benutzung. Ein großes berührungsempfindliches Display, welches spontanen Zugang zu persönlichen Informationen erlaubt.	25
3.3	Schema der Blueboard Frontend- und Backendkomponenten	26
3.4	Schematische Darstellung der Whiteboard-Metapher mit eingebundenen Clientrechnern	27
3.5	Schematische Darstellung der Tabletop-Metapher	29
3.6	Ein Beispiel für die Verwendung von Personal und Group Spaces	30
3.7	Screenshot: BumpTop	32
3.8	Oberfläche der Interface Currents	33
4.1	Alternative Client-Server-Architekturen	37
4.2	Screenshot: Google Maps	39
6.1	Use-Case-Diagramm: Initiativeabhängige Use Cases	49
6.2	Use-Case-Diagramm: Initiativeunabhängige Use Cases	50
6.3	Überblick der Teilsysteme	52
6.4	Oberflächenkonzept	53
6.5	Einteilung der Anwendung in Schichten	56
6.6	Klassendiagramm Magazin	57

1 Einführung

1.1 Kontext und Motivation

Es gibt heutzutage grundsätzlich zwei Arten der Zusammenarbeit von Teams: Das entfernte Arbeiten und das Zusammenkommen zu einem Meeting am gleichen Ort.

Bei der Zusammenarbeit über große Entfernungen hat sich häufig das Internet als Medium der Wahl herausgestellt. Hier bieten sich Möglichkeiten der Verwendung von Voice-over-IP-Diensten oder auch der Einsatz gängiger Instant-Messenger¹ an. Letztere bieten neben der rein textbasierten Kommunikation häufig auch Möglichkeiten zur Sprach- und Videokommunikation und haben sich hier im täglichen Leben als einfache und kostengünstige Alternativen zum klassischen Telefon etabliert. Andere Anwendungen gehen noch einen Schritt weiter: Sie lassen ihre Benutzer gemeinsam an ein und demselben Dokument arbeiten, sofern sie über einen Internetanschluss verfügen.

Die aktuelle Entwicklung bringt eine Erweiterung der Anwendungsmetapher hervor. Neue Anwendungen werden häufig nicht mehr als klassische Desktopanwendungen heruntergeladen, installiert und ausgeführt, sondern laufen direkt im Webbrowser. Beispiele hierfür sind *Meebo*², ein Client für verschiedene Instant-Messaging-Protokolle, oder auch die Instant-Messaging-Funktionen von *Google Mail*³. Weitere Anwendungen wie z. B. die Kalenderanwendung *Google Calendar*⁴ oder auch die Office Produkte von *Ajax13*⁵ zeigen, dass es möglich ist, Desktopanwendungen zentral über einen Webserver zu verteilen und einen Webbrowser hierbei als virtuelle Maschine zu verwenden.

Der Vorteil solcher Anwendungen ist, dass sie von einer zentralen Stelle aus gepflegt und angeboten werden. Ein Administrator muss sich nicht mehr um eine Vielzahl von installierten Client-Versionen kümmern, da die Benutzer bei jedem Zugriff auf die Anwendung immer die aktuelle Version vom Server benutzen.

¹wie z. B. Microsoft Messenger, Skype oder Apple iChat

²Website: <http://www.meebo.com>

³Website: <http://www.google.com/mail>

⁴Website: <http://www.google.com/calendar>

⁵Website: <http://www.ajax13.com>

Im klassischen Büroalltag wird nur selten das volle Potenzial dieser Technologien ausgeschöpft. Die am weitesten verbreiteten Anwendungen sind dabei die Instant-Messaging-Systeme die zur reinen Kommunikation genutzt werden.

Meetings stellen in der heutigen Geschäftswelt einen wichtigen Bestandteil des gegenseitigen Austauschs und der Planung dar. Sie werden sowohl firmenintern mit den eigenen Mitarbeitern oder auch zur Abstimmung mit Kunden oder Partnern eingesetzt. Carola Neumann hat diese Aspekte in ihrer Diplomarbeit (Neumann, 2006) untersucht. Solche Meetings bieten gegenüber der entfernten Kommunikation einige Vorteile. Sie sind häufig persönlicher, da sie den Teilnehmern die Möglichkeit der direkten und natürlichen Kommunikation bieten. Dabei können sie auf Gesichtsmimik, Körperhaltung und Gesten reagieren. Zusätzlich kann man hier auch mit sehr einfachen Mitteln, wie Stift und Papier, gemeinsam Ergebnisse erarbeiten. Es erweist sich allerdings häufig als schwierig, die Ergebnisse eines solchen Meetings in einer Form abzulegen, aus der direkt eine weitere Verarbeitung erfolgen kann. Zu diesem Zweck werden die erarbeiteten Ideen nach der Besprechung häufig nochmals umständlich aufbereitet.

Eine Lösung für dieses Problem wäre z. B. durch den Einsatz technischer Hilfsmittel denkbar. Es existieren hierfür bereits einige Ansätze des Fraunhofer IPSI in Darmstadt, sowie an der Stanford University.

Unternehmen stellen ihren Mitarbeitern in den Büroräumen häufig je ein klassisches Whiteboard zur Verfügung. Diese sind sehr gut geeignet, um schnell Ideen zu entwickeln und diese mit Kollegen zu diskutieren. Es fehlt hierbei allerdings die Möglichkeit, die festgehaltenen Ideen auch dauerhaft zu dokumentieren und an die Gesprächsteilnehmer zu verteilen. Diese Probleme könnten durch den Einsatz digitaler Whiteboards und vergleichbarer Technologie gelöst werden. Für den großflächigen dezentralen Einsatz dieser Technologie sollten allerdings die Folgekosten für Pflege und Wartung möglichst gering bleiben.

Es bietet sich daher eine Kombination aus einer unterstützenden Hardware-Infrastruktur und dem gezielten Einsatz von Webanwendungen an. Hierbei könnten die Vorteile neuer Eingabegeräte zu einer vergleichsweise günstigen Lösung für Unternehmen werden, da sie sowohl die Effektivität von Besprechungen steigern können, als sich auch ihre Kosten bei Wartung und Pflege in Grenzen halten.

Neben Meetings spielt auch die Lösungserarbeitung in kleinen Zweipersonengruppen im täglichen Berufsumfeld eine entscheidende Rolle. Für diese Gespräche bietet sich die bereits genannte Diskussion am Tisch an. Die technologische Unterstützung solcher Prozesse hat hierbei vor allem mit den Herausforderungen einer angepassten Benutzerschnittstelle umzugehen. Diese sollte Initiativwechsel zwischen den Gesprächspartnern unterstützen und einen Wechsel der Perspektive der gemeinsam bearbeiteten Inhalte erlauben.

Diese Punkte beschreiben im wesentlichen das Thema dieser Arbeit.

1.2 Zielsetzung

Im Rahmen dieser Arbeit soll im Einzelnen untersucht werden, inwiefern sich der Einsatz einer auf AJAX basierenden Groupware⁶ eignet. Es soll hierbei ein besonderes Augenmerk auf das gemeinsame Arbeiten mit physisch anwesenden Personen gelegt werden.

Um eine entsprechende Bewertung der Einsetzbarkeit abgeben zu können, wird –begleitend zu den Untersuchungen– eine prototypische Thin-Client-Anwendung für das gemeinsame Arbeiten entwickelt. Es handelt sich hierbei um ein Werkzeug, welches es der Redaktion einer Fotozeitschrift erleichtern soll, Layout-Entscheidungen für eine Ausgabe zu treffen.

Die Auslegung der Arbeit soll hier auf der Zusammenarbeit zweier Personen liegen, die sich an einem Tisch gegenüber sitzen. Es ist diesbezüglich unter anderem mit Herausforderungen der Bereitstellung von Daten, sowie des Wechsels der Initiative zwischen den Bearbeitern umzugehen. Als Beispiel sei hier Abbildung 1.1 gezeigt.



Abbildung 1.1: Beispiel: InteracTable

(Quelle: <http://www.public-accordo.de/wilkhahn/>)

Es wird hierbei Wert auf die Benutzerschnittstelle gelegt. Die Anwendung soll entsprechend der Ideen des User Centered Design und des *Disappearing Computer*⁷ umgesetzt werden. Der Computer als solches soll in den Hintergrund treten und der Benutzer und die Aufgabe in den Vordergrund treten. Es wird hierbei insbesondere eine Ausrichtung auf die spezielle Qualifikation der Benutzer vorgenommen.

⁶genauer: „Pairware“ (vgl. Schrage (1990) zitiert aus (Ishii und Kobayashi, 1992, S. 526))

⁷s. Weiser (1991) und Russell u. a. (2005)

Die Anwendung soll vollständig in einem Webbrowser ausgeführt werden. Als technologischer Hintergrund soll AJAX dienen.

1.3 Gliederung

Kapitel 2 stellt Infrastrukturen für Collaborative Workspaces vor. Es werden hier spezielle Hardware- und Raumkonzepte erläutert, mit deren Hilfe sich ein Collaborative Workspace aufbauen lässt.

In Kapitel 3 wird darauf aufbauend gezeigt, wie sich ein Collaborative Workspace nutzen lässt. Es werden hier Beispiele für Einsatzkontexte gegeben, in denen die im vorhergehenden Kapitel vorgestellten Technologien verwendet werden.

Kapitel 4 schafft die entsprechenden Grundlagen zur AJAX-Technologie. Es soll hier ein Überblick zu bestehenden Anwendungen gegeben werden, die im Bezug zu dem in dieser Arbeit angestrebten Ziel stehen.

Das Kapitel 5 beschreibt ein fiktives Beispielszenario "Virtueller Leuchttisch zur Erstellung einer Fotozeitschrift". Es soll hier die Grundlage für die weitere Untersuchung der Fragestellung an Hand eines Beispiels geschaffen werden.

Der in Kapitel 6 vorgestellte Systementwurf wird basierend auf dem Szenario erstellt. Es wird zunächst die Architekturvision eines denkbaren Gesamtsystems vermittelt, um daraus dann die Anforderungen für die Realisierung eines explorativen Prototypen abzuleiten.

Kapitel 7 gibt eine Zusammenfassung und Bewertung der in der Arbeit dargestellten Themen ab. Anschließend wird hier ein Ausblick auf die weitere Entwicklung gewagt.

2 Infrastrukturen für Collaborative Workspaces

Das kooperierende Arbeiten in einer Face-to-Face-Umgebung hat viele Vorteile. Die Teilnehmer können hier in einer gewohnten Weise miteinander kommunizieren. Unterschwellige Botschaften über Mimik und Körperhaltung, sowie über Gesten sind sichtbar und die Kommunikationspartner können so besser und natürlicher aufeinander eingehen. Diese Vorteile werden ausführlich in [Hirrichs \(2005\)](#) beschrieben.

In [Neumann \(2006\)](#) werden verschiedene Szenarien innerhalb eines Projektverlaufs betrachtet. Die Arbeit konzentriert sich auf die zentralen Kommunikationsprozesse der heutigen Arbeitswelt, wie Meetings, Diskussionen, Präsentationen und Projektarbeiten. Sie zeigt auf, wie das Erarbeiten von Inhalten und Ideen, das Protokollieren von Ergebnissen und die spätere Verteilung auf effiziente Weise technologiegestützt ablaufen kann.

Die Idee eines *Collaborative Workspace* versucht eben diese Gedanken in einer technologiegestützten Umgebung umzusetzen, so dass sich die verschiedenen Angebote möglichst gut ergänzen. Es wird ebenfalls versucht, die Ideen des Ubiquitous Computing umzusetzen, welche unter anderem von Terry Winograd und Fernando Flores ([Winograd und Flores, 1986](#)) geprägt wurden. Computer sollen unterstützen, aber selbst kaum wahrgenommen werden.

Erreicht wird dieser Zustand durch die Verwendung von Benutzerschnittstellen, die eine intuitive Bedienung erlauben.

Im Folgenden sollen beispielhaft zwei Infrastrukturen, die den Aufbau eines Collaborative Workspace zum Ziel haben, vorgestellt werden. Es handelt sich hierbei um das *Roomware*-Projekt vom Fraunhofer IPSI in Darmstadt, sowie das *iRoom*-Projekt des HCI in Stanford. Eine nähere Erläuterung von Beispielen zu Einsatzkontexten folgt im Kapitel 3.

2.1 Roomware

Das Roomware-Projekt wird vom Fraunhofer IPSI in Darmstadt entwickelt und ist ein Teil des *i-Land*-Projekts. Es wurden im Rahmen des Projektes einige grundlegende Komponenten

entwickelt, die das Arbeiten in einer kollaborativen Umgebung ermöglichen sollen. Die Einzelteile werden in der Dissertation von Peter Tandler ([Tandler, 2004](#)) beschrieben. Die Abbildung 2.1 zeigt eine Übersicht der Roomware-Komponenten. Die Abgebildeten Komponenten werden im folgenden kurz beschrieben.



Abbildung 2.1: Übersicht der Roomware-Komponenten
(von vorne nach hinten: ConnectTable, InteracTable, Commchair und DynaWall)
(Quelle: [Tandler \(2004\)](#), S. 66)

2.1.1 DynaWall

Die *DynaWall* ist ein Wanddisplay mit einer Breite von 4,50 m und einer Höhe von 1,10 m. Sie setzt sich aus drei berührungsempfindlichen Einzeldisplays zusammen, deren Ausgaben zu einem Bild synchronisiert werden. Die Bereitstellung einer solchen Displayfläche soll Gruppen, wie z. B. Projektteams, die Möglichkeit gegeben, große Informationsstrukturen darzustellen und diese von zwei oder mehr Personen bearbeiten zu lassen. Diese Art der Zusammenarbeit ist in Abbildung 2.2 beispielhaft dargestellt.

2.1.2 CommChair

Der *CommChair* (vgl. Abbildung 2.3) ist ein Stuhl mit einem integrierten Computer. Dieser wird über ein Display bedient, auf welchem über einen Stift Eingaben gemacht werden kön-



Abbildung 2.2: Zwei Personen an einer DynaWall
(Quelle: [Tandler \(2004\)](#), S. 17)

nen. Der *CommChair* verfügt über eine kabellose Netzwerkschnittstelle und eine integrierte Stromversorgung, um eine größtmögliche Mobilität zu gewährleisten.

Der *CommChair* soll sowohl die Einzelarbeit, als auch das Arbeiten in der Gruppe erleichtern, indem er z. B. in Kombination mit der *DynaWall* eingesetzt wird.

2.1.3 InteracTable

Der *InteracTable* ist ein Tisch mit eingelassenem berührungsempfindlichem Bildschirm, der gleichzeitig von mehreren Personen benutzt werden kann.

Die Besonderheit beim Einsatz eines horizontal gelagerten Bildschirms besteht darin, dass es für die umstehenden Anwender keine objektive Definition von *oben*, *unten*, *links* und *rechts* gibt. Dieser Umstand erschwert den Einsatz von Standardsoftwareprodukten, die nicht speziell auf diese Situation ausgelegt wurden.



Abbildung 2.3: Die zweite Generation des *CommChair*
(Quelle: [Tandler \(2004\)](#), S. 17)

2.1.4 ConnecTable

Der *ConnecTable* ist ein freistehender Rechner, der ein schwenkbares berührungsempfindliches Display besitzt. Das besondere am *ConnecTable* stellt seine Fähigkeit dar, sich spontan mit der Displayfläche eines zweiten *ConnecTable* zu verbinden, um so einen schnellen Austausch oder eine Besprechung zwischen zwei Personen zu gewährleisten. Technisch wird dieses Verhalten über RFID-Tags und -Empfänger gelöst, die eine Verbindung der Displayflächen einleiten, sobald sich zwei Tische nahe genug sind. Die beiden Benutzer können sich so spontan für eine kurze Besprechung einen Bildschirm teilen und ihre Ideen austauschen. Die Software des *ConnecTable* sorgt dafür, dass die ausgetauschten Daten nach der Trennung auch auf beiden Geräten vorhanden sind. Der Datenaustausch zwischen den Geräten geschieht über eine drahtlose Netzwerkverbindung.

2.1.5 Zusammenfassung

Die vorgestellten Komponenten bieten jeweils einzeln einen gewissen Mehrwert für den Benutzer. Da sie meist über eine mobile Stromversorgung verfügen und sich in ein drahtloses Netzwerk integrieren lassen, stellen sie kaum Anforderungen an ihre Umgebung und sind sehr vielseitig einsetzbar.

Die Komponenten des Roomware-Projekts entfalten ihre volle Leistungsfähigkeit allerdings



Abbildung 2.4: Fraunhofer InteracTable
(Quelle: [Fraunhofer-IPSI \(2005\)](#))



Abbildung 2.5: Zwei miteinander verbundene ConnectTables
(Quelle: [Tandler \(2004\)](#), S. 163)

erst, wenn sie sinnvoll kombiniert werden. So lassen sich beispielsweise mehrere Comm-Chairs oder ConnectTables auch in Verbindung mit der *DynaWall* einsetzen. Ein entsprechendes Beispiel ist in Kapitel 3.2 beschrieben.

2.2 iRoom

Der *iRoom*¹ ist als Teil des *interactive Workspaces*-Projekts an der Stanford University entstanden. Er soll eine praktische Anwendung der Überlegungen zum Ubiquitous Computing liefern. Bradley Johanson beschreibt in [Johanson \(2002\)](#) den *iRoom* und seine Komponenten.



Abbildung 2.6: Der *iRoom* in Benutzung
(Quelle: [Johanson \(2002\)](#), S. 5)

Die Abbildung 2.6 zeigt den Raum in Benutzung, eine schematische Darstellung der einzelnen Komponenten findet sich in der Abbildung 2.7. In der Mitte des Raumes befindet sich ein großer Konferenztisch, der so genannte *iTable*, mit einem integrierten Bildschirm. An der langen Seite des Tisches findet sich eine horizontale Anordnung von drei Smartboards und

¹Kurzform für *Interactive Room*

an der anderen Wand ist ein hochauflösender Bildschirm angebracht, welcher als *Information Mural* bezeichnet wird. Diese stellt ein aus zwölf Einzelbildern² zusammengesetztes Bild dar, welches in einem 32 PC Rendering-Cluster errechnet wird. Es besteht zusätzlich für die Benutzer die Möglichkeit, zu diesen fest installierten Geräten noch eigene Geräte in die Umgebung des Raums via WLAN einzubinden.

Einige dieser Komponenten weisen Ähnlichkeiten zu den bereits in Kapitel 2.1 beschriebenen Roomware-Komponenten auf.

Der *iTable* ähnelt in seiner Funktion sehr dem *InteracTable*. Er bietet ebenfalls einen integrierten Bildschirm, der es den Personen am Tisch ermöglicht, gemeinsam an einer Aufgabe zu arbeiten.

Der *iRoom* weist auch eine horizontale Anordnung von drei Smartboards auf, die Funktionalität der vergleichbaren *DynaWall* findet sich allerdings eher in der *Information Mural* wieder.

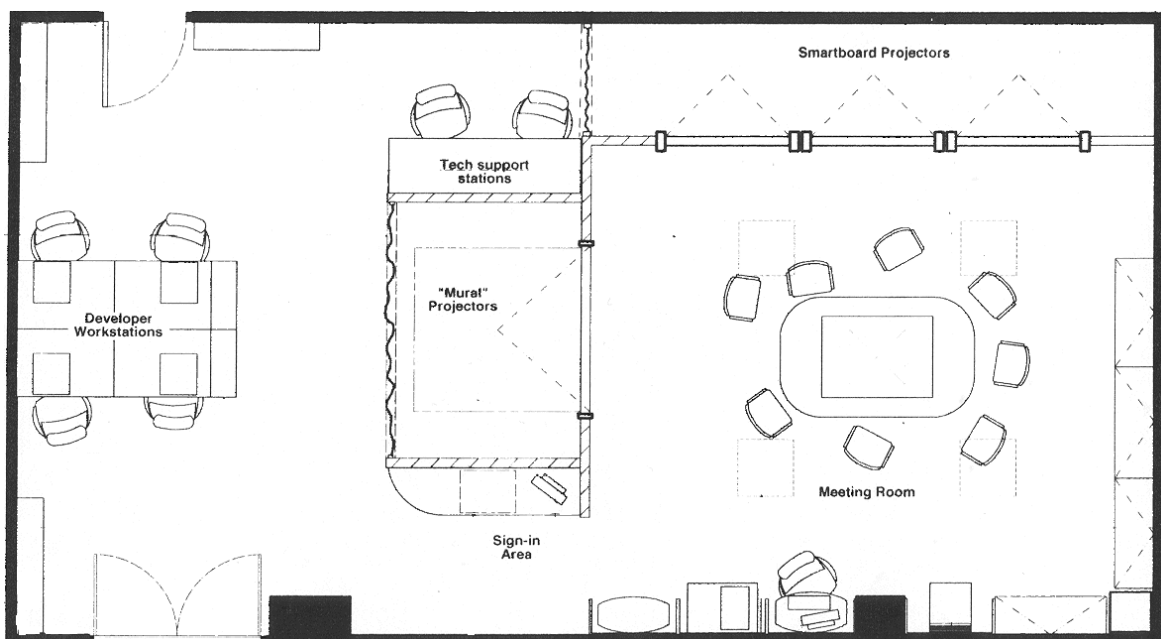


Abbildung 2.7: Schematische Darstellung des *iRoom*
(Quelle: Johanson (2002), S. 8)

Eine wichtige Idee des *iRoom* ist, dass er möglichst viel Bildschirmfläche zur Verfügung stellen soll. Johanson stellt in seiner Dissertation heraus, dass dieses für das gemeinsame Arbeiten sehr hilfreich ist, da es die Gruppenmitglieder in die Lage versetzt, einen Blick auf das vollständige Problem zu haben (vgl. Johanson (2002), S. 6).

²Anordnung: 4 x 3

2.3 Der Collaborative Workspace an der Hochschule für Angewandte Wissenschaften Hamburg

An der Hochschule für Angewandte Wissenschaften Hamburg wird derzeit ebenfalls ein solcher Raum eingerichtet. Es sollen in diesem Umfeld Untersuchungen unternommen werden, inwiefern sich die Erkenntnisse des computergestützten kooperierenden Arbeitens in praktische Umsetzungen überführen lassen.

Es existieren bereits eine Reihe von Abschlussarbeiten in diesem Umfeld. Einige einführende Beschreibungen finden sich in den Diplomarbeiten von Carola Neumann ([Neumann, 2006](#)), Roman Bartnik ([Bartnik, 2006](#)) und Lars Burfeindt ([Burfeindt, 2006](#)).

2.4 Zusammenfassung

Die vorgestellten Umgebungen haben alle das Ziel, das kooperierende Arbeiten am selben Ort zu fördern. Sie gehen hierfür teilweise sehr ähnliche Wege, indem sie auf dem Gedanken des Ubiquitous Computing aufbauen. Es wird somit in beiden Umgebungen die natürliche Arbeitsweise von Gruppen unterstützt.

Der auffälligste Unterschied zwischen den Ansätzen ist, dass sich die Lösung vom Fraunhofer IPSI eher als eine Art Baukasten zeigt. Es ist hier möglich, die verschiedenen Elemente frei nach Bedarf zu kombinieren. Der *iRoom* stellt hingegen eine Gesamtlösung für einen Konferenzraum bereit.

3 Einsatzkontexte eines Collaborative Workspace

Nachdem im vorhergehenden Abschnitt Beispiele für die Infrastruktur eines Collaborative Workplace gegeben wurden, sollen in diesem Abschnitt ergänzend drei Beispiele für Einsatzkontexte unter Verwendung der vorgestellten Komponenten aufgezeigt werden. Es handelt sich hierbei um drei Beispiele, die mit einer für den jeweiligen Fall typischen Architektur verbunden sind.

Das Ziel ist hierbei, immer ein Maximum an Nutzen aus der jeweiligen Konfiguration zu ziehen. Zum einen, um möglichst effiziente Entscheidungen treffen zu können, zum anderen aber auch, um eine hohe Qualität von Dokumentation oder weiterverwendbaren Ergebnissen zu erhalten.

3.1 Digitales Whiteboard

Der Nutzungskontext *digitales Whiteboard* setzt einen vertikal ausgerichteten Bildschirm voraus, der ebenfalls auch für Eingaben verwendet werden kann, um die Eigenschaften einer herkömmlichen Wandtafel nachzubilden. Hierbei kann es sich z. B. um die im Kapitel 2.1.1 vorgestellte *DynaWall* oder auch die im *iRoom* installierten Smartboards (s. Kapitel 2.2) handeln. In der Abbildung 3.1 ist der Aufbau eines solchen Systems schematisch dargestellt. Es befinden sich hier mehrere Benutzer vor dem Display, welche auf diesem Eingaben tätigen und die Aktionen der übrigen Teilnehmer wahrnehmen und auch selbst wahrgenommen werden.

Ein ähnlicher Aufbau ist beispielsweise in [Ishii und Kobayashi \(1992\)](#) beschrieben. Die Autoren weisen auch auf Nachteile hin, die aus der fehlenden Trennung der Arbeitsbereiche der Benutzer resultieren¹. In [Tandler \(2004\)](#) wird diese Problematik allerdings noch etwas weiter präzisiert:

¹vgl. [Ishii und Kobayashi \(1992\)](#), S. 527

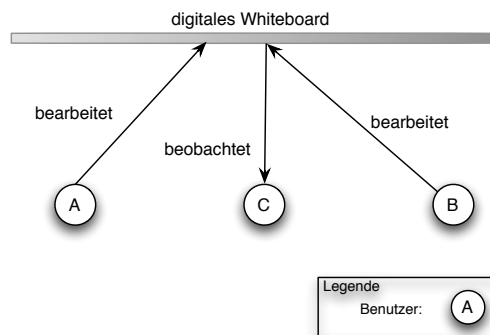


Abbildung 3.1: Schematische Darstellung der Whiteboard-Metapher

Large interactive surfaces raise the need for new forms of interaction, as it is, e.g., not possible (or convenient) to easily reach all objects on the wall

*Peter Tandler
(Tandler, 2004, S. 17)*

Er beschreibt hier, dass diese spezielle Form der Bereitstellung eines kombinierten Ein- und Ausgabemediums auch spezielle Formen der bereitgestellten Software benötigen. Der bedeutendste Unterschied zwischen den Überlegungen von [Tandler](#) und [Ishii und Kobayashi](#) liegt wohl in der Größe der bereitgestellten Displayfläche. Während das vorgestellte *ClearBoard* ([Ishii und Kobayashi, 1992](#)) zwar auch eine große Fläche bietet, so ist die von der *DynaWall* bereitgestellten Bildschirmfläche erheblich größer, da sie bereits in der Breite mehrere Metern umfasst.

Mögliche Einsatzszenarien für die von [Tandler](#) vorgestellte *DynaWall* wären z. B. die Besprechung eines Projektplans oder die Nebeneinanderstellung ähnlicher Informationen zur Klärung eines Sachverhalts. Eine der beteiligten Personen kann hier Daten zur Diskussion stellen, die von den übrigen Teilnehmern aufgegriffen und weiterentwickelt werden können. Der von [Ishii und Kobayashi](#) herausgestellte Nachteil der fehlenden Trennung der Arbeitsbereiche wäre somit aufgehoben und würde somit eher zu einem Vorteil werden.

Die Anwendung würde sich in diesem Kontext als eine klassische Einzelplatzanwendung präsentieren. Zwar ist es denkbar und in der Realisierung sehr wahrscheinlich, dass die Darstellung der Displayfläche, die Bereitstellung der Daten sowie Werkzeuge auf mehrere Rechner aufgeteilt werden, den Benutzern würden diese jedoch verborgen bleiben, da sie lediglich die Funktionen auf dem Display verwenden.

3.1.1 Beispiel: BlueBoard (IBM)

Mit *BlueBoard* bezeichnen [Russell u. a. \(2002\)](#) große berührungsempfindliche Bildschirme, die es autorisierten Benutzern erlauben, in eine spontane Interaktion miteinander zu treten. Diese Bildschirme werden an öffentlichen Orten aufgestellt und erlauben einzelnen Personen oder auch kleinen Gruppen einen Informationsaustausch, bzw. das Präsentieren oder Entwickeln von Ideen. Ein Benutzer meldet sich am *BlueBoard* mit Hilfe eines RFID-Tags an, das er auf ein entsprechendes Lesegerät am *BlueBoard* legt.



Abbildung 3.2: Das Blueboard in Benutzung. Ein großes berührungsempfindliches Display, welches spontanen Zugang zu persönlichen Informationen erlaubt.
(Quelle: [Russell u. a. \(2002\)](#), S. 2)

Die Idee des *BlueBoards* lebt von der spontanen Verwendung des Geräts:

In ordinary use, the BlueBoard is intended for both very fast personal use (walk up, check your calendar, walk away – all within 5 seconds) [...]

*Russell u. a.
(Russell u. a., 2002, S. 2)*

Abbildung 3.2 zeigt eine solche Situation, in der sich ein Mitarbeiter am *BlueBoard* angemeldet hat und seinen persönlichen Kalender überprüft.

Das *BlueBoard* selbst ist ein Thin-Client, auf welchem die Benutzerdaten überprüft werden und von welchem aus die dargestellten Inhalte angefordert werden.

Bei der Anmeldung über das Lesegerät sendet der *BlueBoard*-Client die Benutzer-ID an einen Server. Dieser sendet dann eine URL zurück, welche auf die persönlichen Inhalte des Benutzers zeigt. Der Ablauf zeigt sich in der Abbildung 3.3.

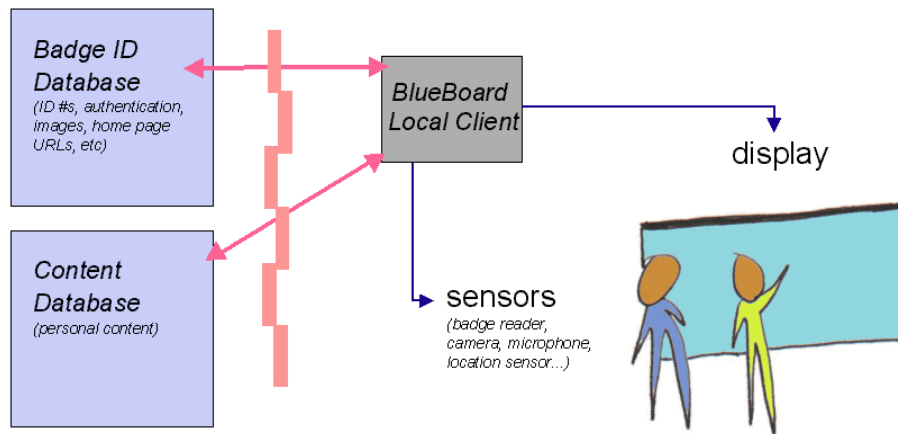


Abbildung 3.3: Schema der Blueboard Frontend- und Backendkomponenten
(Quelle: Russell u. a. (2002), S. 3)

3.2 Digitales Whiteboard in Kombination mit privaten Clients

Der im vorhergegangenen Abschnitt beschriebene Aufbau erlaubt es seinen Benutzern lediglich, Informationen auf einem öffentlichen Bildschirm darzustellen. Es sind allerdings auch Anwendungen wie z. B. ein moderiertes Brainstorming oder ein Einsatzleitstand denkbar, in denen es sinnvoll ist, Informationen zunächst in einer privaten Umgebung vorzubereiten, bevor diese den übrigen Teilnehmern präsentiert werden. Ein entsprechender Aufbau ist in der Abbildung 3.4 dargestellt. Der große öffentliche Bildschirm wird von einem Moderator bedient. Die übrigen Gruppenteilnehmer benutzen ihre Rechner zur Aufbereitung von Inhalten und können diese dann auf dem öffentlichen Bildschirm bereitstellen.

3.2.1 Beispiel: Moderiertes Brainstorming

In einem moderierten Brainstorming kommt eine Gruppe von Personen zusammen. Jede dieser Personen hat ein eigenes Notebook, auf dem die Clientsoftware für das Vorbereiten von Inhalten, sowie den Zugriff auf den öffentlichen Bildschirm bereitsteht. Es ist zudem

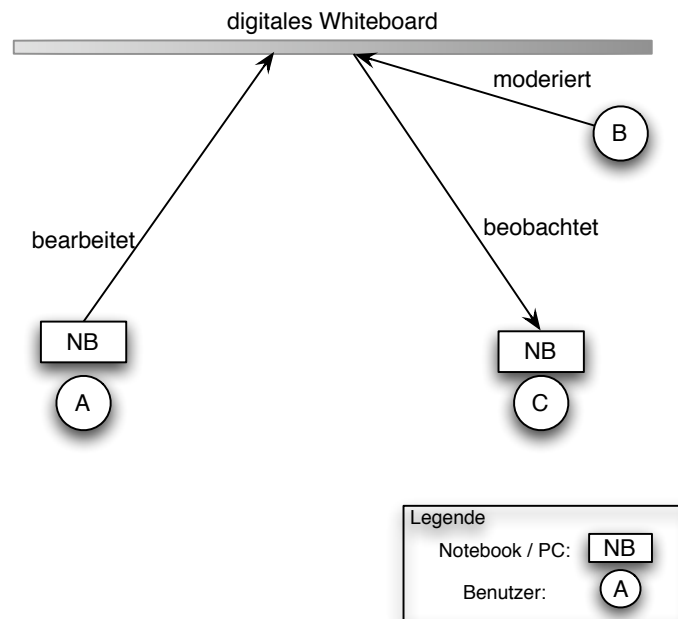


Abbildung 3.4: Schematische Darstellung der Whiteboard-Metapher mit eingebundenen Clientrechnern

denkbar, dass die Clients Zugriff auf Netzwerkressourcen wie z. B. ein Firmennetzwerk oder das Internet haben.

Sobald ein Inhalt soweit aufbereitet wurde, dass er den übrigen Teilnehmern präsentiert werden kann, wird er auf den öffentlichen Bildschirm *geschoben* und kann mit den übrigen Teilnehmern diskutiert werden. Sollten noch weitere Bearbeitungsschritte notwendig sein, die nicht die Aufmerksamkeit der gesamten Gruppe erfordern, so können die Teilnehmer sich Inhalte vom öffentlichen Bildschirm auf ihren persönlichen Arbeitsplatz ziehen und dort anpassen.

Es wird in dieser Umgebung ein Moderator eingesetzt, um die Geschehnisse auf dem öffentlichen Bildschirm zu steuern. Er könnte beispielsweise Freigabewünsche für neue Inhalte entsprechen oder diese ablehnen, um somit eine geordnete Diskussion zu ermöglichen.

3.2.2 Beispiel: Einsatzleitstand

Der im ersten Beispiel beschriebene Aufbau lässt sich nahezu unverändert auch auf den Kontext eines stark vereinfachten Einsatzleitstandes z. B. einer Feuerwehr oder eines Sicherheitsdienstes übertragen. In diesem Zusammenhang könnte jeder angeschlossene Client

einen eigenen Bereich überwachen und kritische Informationen den übrigen Teilnehmern in aufbereiteter Form auf dem öffentlichen Bildschirm bereitstellen.

Die Benutzer an den Clients würden somit eine filternde Funktion auf die von außen kommenden Daten ausüben. Der Vorteil gegenüber der Verwendung von isolierten Clients ist, dass wichtige Daten schnell allen Teilnehmern bekannt gemacht werden können. Ein anderes Extrem wäre die ständige Darstellung aller Daten auf einem öffentlichen Bildschirm. Dieses würde jedoch bedeuten, dass wahrscheinlich auch viele irrelevante Daten dargestellt werden, die von wichtigen Ereignissen ablenken können.

3.2.3 Bedeutung für die Architektur

Die vorliegende Architektur entspricht einer Client-/Server-Architektur. Sie wird von den Benutzern auch als solche wahrgenommen werden, da die Arbeit hier physisch auf mehrere Geräte verteilt stattfindet. Je nach angestrebtem Umfeld können die Clients fest in das System eingebunden sein² oder sich spontan in dieses integrieren³.

3.3 Tabletop

Unter der *Tabletop*-Metapher versteht man einen Aufbau mit einem Bildschirm, der wie ein Tisch horizontal ausgerichtet ist. Als Beispiele für eine entsprechende Hardware wurden bereits der *iTable* oder der *InteracTable* vorgestellt.

In [Kruger u. a. \(2003\)](#) wird dargelegt, dass das Arbeiten mit Stift und Papier an einem Tisch noch immer eine häufig angewandte Art Kooperation zur Ergebnisbindung ist. Zwar stellen ihren Aussagen nach Desktopcomputer häufig einen besseren Zugang zu Informationen bereit, doch würden diese während der Gruppenarbeit die Anwender zwingen sich in eine unangenehme Nähe zueinander zu begeben. Die Gruppe wird gezwungen sich nah beieinander vor einen Bildschirm zu setzen, sofern sie alle den gleichen Blick auf die Arbeitsfläche⁴ haben wollen. Zusätzlich würden sie sich so in einer ungewohnten Form der Interaktion unterziehen, welche die natürliche Kommunikation erschwert. Bei der Arbeit am Tisch ist es den Gruppenteilnehmern möglich sich um diesen herum zu verteilen. Es ist jedem möglich, den Anderen anzuschauen und seine Handlungen zu beobachten⁵.

²vgl. Leitstand

³vgl. Brainstorming

⁴den Monitor

⁵s. a. [Scott u. a. \(2003\)](#)

Der Einsatz eines solchen Tisches bietet gute Möglichkeiten von den Nutzern als ein bekanntes Medium angenommen zu werden. Die Herausforderungen bestehen hier darin, die Anwendungen entsprechend der Zielgruppe und der Aufgabenstellung zu gestalten. Einen wichtigen Faktor stellt in diesem Zusammenhang das Userinterface dar.

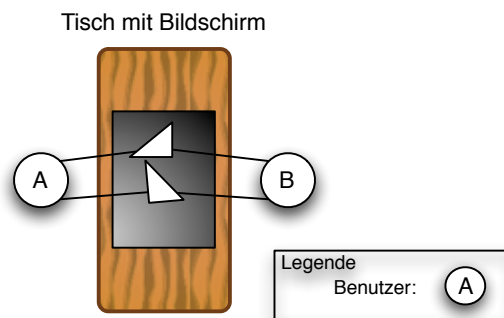


Abbildung 3.5: Schematische Darstellung der Tabletop-Metapher

3.3.1 Besonderheiten der Oberfläche

Eine optimale Verteilung der Benutzer wird sein, dass sich diese nach Möglichkeit gegenüber, bzw. sich so verteilen, dass sie möglichst viel Platz für eigene Materialien auf dem Tisch haben. Scott u. a. (2003) beschreiben einige Formen der Verteilung um einen Tisch. Es wird dabei darauf hingewiesen, dass die Gruppenteilnehmer sich für kommunikationsintensive Arbeiten so verteilen, dass sie sich gegenüber oder auch um die Ecke des Tisches herum sitzen. Dieses Die Abbildung 3.5 zeigt eben diese Anordnung. Die Besonderheit hierbei ist, dass es auf diese Weise für keinen der Benutzer ein objektives *Oben* oder *Unten* gibt.

Kruger u. a. (2003) haben in ihren Untersuchungen festgestellt, dass Teilnehmer in solchen Situationen dazu neigen, sich eigene Bereiche, so genannte *Personal Spaces* einzurichten, die dazu dienen, Teilprobleme zu lösen, welche nicht die Aufmerksamkeit aller Bearbeiter benötigen. Zusätzlich gibt es einen *Group Space*⁶, in dem die Einzelergebnisse zusammengetragen werden. Dieser gemeinsame Arbeitsbereich kann immer der Person zugedreht werden, welche derzeit die Initiative besitzt. Ein Initiativewechsel kann durchgeführt werden, indem ein Gruppenteilnehmer diese *ergreift* oder diese explizit an einen anderen *abgibt*. Der zweite Fall ist insbesondere dann sinnvoll, wenn über den aktuellen Zwischenstand diskutiert werden soll.

⁶In der Abbildung als *Group Space - puzzle* bezeichnet

Die Abbildung 3.6 zeigt eine solche Anordnung. Zwei Personen versuchen hier gemeinsam ein stark orientierungsabhängiges Puzzle⁷ zu lösen. Hier ist zusätzlich noch ein *Group Space - preview* abgebildet, welcher den beiden Bearbeitern als Hilfestellung für das zu lösende Puzzle gilt.

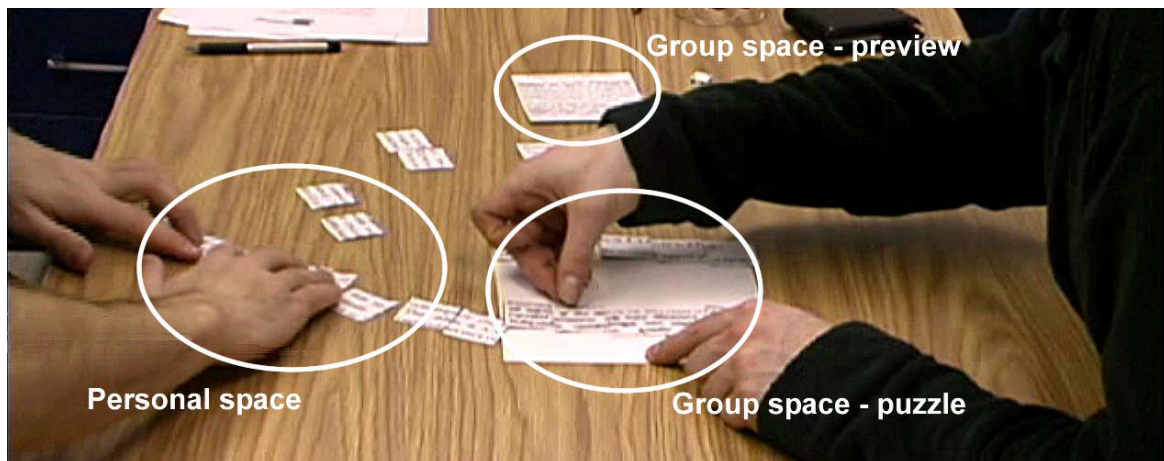


Abbildung 3.6: Ein Beispiel für die Verwendung von Personal und Group Spaces
(Quelle: Kruger u. a. (2003), S. 372)

Eine Tabletop-Anwendung sollte daher in der Lage sein, den verwendeten *Group Space* jeweils dem Bearbeiter zuzuwenden, der die Initiative besitzt. Von Kruger u. a. werden hierzu einige Konzepte vorgestellt.

Fixed orientation Der dargestellte Bildausschnitt besitzt eine feste Orientierung. Die Benutzer sitzen nebeneinander an einer Seite des Tisches.

Manual orientation Die Benutzer drehen die entsprechenden Teile des Bildes manuell. Dieses entspricht am ehesten der gewohnten Arbeit an einem Tisch mit Stift und Papier.

Person-based automatic orientation Bei diesem Ansatz werden die Inhalte immer zu dem Benutzer ausgerichtet, der sie bearbeitet. Es wird dabei angenommen, dass dieser auch den größten Nutzen von einem solchen Verhalten hat.

Environment-based automatic orientation Es wurde beim *Person-based automatic orientation* erkannt, dass es schwierig ist, immer herauszufinden wo sich die bearbeitende Person befindet. Es kann somit passieren, dass ein Element auf eine falsche Seite des Tisches ausgerichtet wird. Es kam auch die Frage auf, ob eine Ausrichtung auf den Bearbeiter immer sinnvoll ist, da manchmal auch *über Kopf* gearbeitet wird, um anderen besser etwas zeigen zu können. Es wurde somit dieser vierte Ansatz identifiziert, der die Elemente –entsprechend ihrer Position auf dem Tisch– ausrichtet.

⁷In diesem Fall: Ein Textpuzzle

Eine weitere Herausforderung in diesem Zusammenhang stellt das Weitergeben von Informationen dar. Die mit der Aufgabe betrauten Personen sollten in der Lage sein, Teilergebnisse oder einzelne Informationen miteinander auszutauschen, ohne dass jemand aufstehen muss um etwas zu bringen oder abzuholen.

3.3.2 Beispiel: Physikbasierte Interaktion

Bei physikbasierter Interaktion handelt es sich um die Simulation physikalischer Vorgänge bei der Arbeit an einem Tabletop-Bildschirm. Sie ermöglicht den Benutzern sich gegenseitig Objekte, welche auf dem Tisch dargestellt werden, zuzuspielen.

Die Umgebung beinhaltet eine physikalische Simulation, die es ermöglicht, Elementen ein natürliches Verhalten bezüglich ihrer Form und Größe zu geben. Es sind somit Interaktionen möglich, die ein gleichzeitiges Verlangsamen der Bewegung sowie die Drehung des Objekts erlauben. Ähnliche Gedanken wurden bereits von [Streitz u. a. \(1999\)](#) gemacht. Diese bezogen sich allerdings auf das Verschieben von Elementen auf einem großen Wandbildschirm.

Der Einsatz von physikbasierten Interaktionsmöglichkeiten wurde in [Agarawala \(2006\)](#) untersucht. Ziel der Arbeit war es, eine neue Form des Computerdesktops, den so genannten *BumpTop*⁸, zu schaffen. Dieser soll sich mehr an den natürlichen Gegebenheiten der Umwelt orientieren und dem Benutzer somit ein vertrauterer und effizienteres Arbeiten am Computer ermöglichen.

Die Ausrichtung des *BumpTop* ist auf die Verwendung als Einzelplatzanwendung auf einem Tablet-PC mit Stifteingabe ausgerichtet. Es wurde jedoch im vergangenen Semester im Projekt Anwendungen des Masterstudiengangs an der Hochschule für Angewandte Wissenschaften Hamburg ein Prototyp erstellt, der ebenfalls basierend auf einer Simulation physikalischer Eigenschaften das kooperierende Arbeiten an einem Tisch ermöglicht. Eine Beschreibung findet sich in [Roßberger \(2007\)](#).

3.3.3 Beispiel: Interface Currents (Uta Hinrichs)

Uta Hinrichs hat in ihrer Diplomarbeit ([Hinrichs, 2006](#)) und in diversen weiteren Veröffentlichungen⁹ ein Konzept für das gemeinsame Arbeiten an einem großen Tabletop-Display dargelegt.

⁸s. Abbildung 3.7

⁹s. [Hinrichs \(2005\)](#), [Hinrichs u. a. \(2005b\)](#), [Hinrichs u. a. \(2005a\)](#) und [Hinrichs u. a. \(2006\)](#)



Abbildung 3.7: Screenshot: BumpTop
(Quelle: [Agarawala \(2006\)](#), S. 45)

Sie beschreibt eine Art Umlauf von Elementen¹⁰, die am äußeren Rand des Bildschirms, bzw. auf festgelegten Kanälen *fließen*. Es ist somit für den Benutzer nicht erforderlich, dass dieser über den Tisch greifen muss, um ein Element zu bekommen. Diese fließen ihm einfach zu. Es kann somit die gesamte mittlere Fläche für das gemeinsame Arbeiten verwendet werden, da keine Elemente die Mitte kreuzen um hin- und hergereicht zu werden. Die Benutzer werden ebenfalls nicht gezwungen, sich unbequem über den Tisch zu beugen oder gar aufzustehen, um an bestimmte Elemente zu gelangen.

Die Abbildung 3.8 zeigt die Anordnung auf einem solchen Tischdisplay. Die Benutzer können sich Bilder aus dem Umfluss greifen und in der Mitte platzieren.

Als Nachteil dieser Art der Verteilung von Arbeitsmaterialien lässt sich anführen, dass es einige Zeit in Anspruch nehmen kann, bis ein Element seinen Weg zum Empfänger hinter sich gebracht hat. Es kann zusätzlich schwierig sein, dass gewünschte Element *im Auge* zu behalten, bis es in Reichweite ist. Dieser Umstand könnte sich als ausgesprochen kontraproduktiv erweisen, da das Gruppenmitglied, im schlimmsten Fall für die Zeit, die es auf das Element wartet, von der Arbeit abgehalten wird.

¹⁰hier: Bilder



Abbildung 3.8: Oberfläche der Interface Currents
(Quelle: [Hinrichs \(2006\)](#), S. 86)

3.3.4 Bedeutung für die Architektur

Die Überlegungen für die technische Architektur sind hier größtenteils identisch mit der im Abschnitt 3.2.3 beschriebenen Architektur. Unterschiede bestehen hier darin, dass es sich hier lediglich um einen anzusteuern den Bildschirm handelt, wodurch sich die Displayansteuerung einfacher realisieren lässt.

3.4 Zusammenfassung

In diesem Kapitel wurden einige Einsatzkontexte für die Verwendung der im Kapitel 2 vorgestellten Infrastrukturen aufgezeigt. Anhand der Beispiele wurde gezeigt, dass jeder dieser Kontexte eigene Anforderungen an die Gestaltung der Architektur und der Benutzerschnittstelle hat.

Im Falle der Benutzerschnittstelle ist das insbesondere bei Tabletop-Umgebungen der Fall. Hier reicht eine leichte Modifikation bereits vorhandener Desktopanwendungen nicht aus. Was bei der Verwendung eines vertikalen Bildschirms zumindest als möglich erscheint, wird hier auf Grund der fehlenden objektiven Ausrichtung des Bildes zusätzlich erschwert. Zwar wäre eine (simulierte) physische Drehung des gesamten Bildschirminhalts denkbar, dieses

würde allerdings immer einen großen Teil der Gruppenteilnehmer von der Arbeit ausschließen.

3.5 Bewertung der vorgestellten Ansätze

Die vorgestellten Ansätze bieten alle eine auf ihren jeweiligen Anwendungskontext zugeschnittene Architektur. So ist es für fest installierte integrierte Umgebungen mit einem fest umrissenen Anwendungskontext sinnvoll, die Anwendung auf den Clients zu verteilen und auch auszuführen. Als Beispiel wäre hier der Einsatzleitstand aus Abschnitt 3.2.2 zu erwähnen.

Eine andere Situation findet sich bei dem in Abschnitt 3.1.1 beschriebenen Szenario wieder. Hier wird den Benutzern bewusst die Freiheit eingeräumt, sich ihre Oberfläche selbst nach ihren Anforderungen zu gestalten. Es handelt sich um einen sehr leichtgewichtigen Ansatz. Die Vorteile liegen in der einfacheren Handhabbarkeit von Konfigurationen und Versionen der Software. Das in Kapitel 3.1.1 vorgestellte *BlueBoard* soll an vielen Orten zu spontaner Interaktion bereit stehen. Es ist daher sinnvoll, die Client-Software möglichst einfach zu halten, da dieses Aufwand bei der Wartung und Pflege der Software einspart.

4 AJAX-Konzepte

AJAX ist eine Sammlung von Technologien aus dem WEB2.0-Umfeld. Es sollen in diesem Kapitel zunächst die Hintergründe und Ideen des WEB2.0 erläutert werden, um dann darauf aufbauend eine Beschreibung der AJAX-Technologie geben zu können. Abschließend werden in diesem Kapitel einige Beispiele bestehender AJAX-Anwendungen gezeigt.

4.1 Hintergründe

Der Begriff WEB2.0 wurde von Tim O'Reilly geprägt. Die Eckpunkte seiner Ideen¹ werden im folgenden kurz dargestellt.

Das Internet als Plattform. Das Internet wird im WEB2.0 nicht mehr als Medium, sondern als Plattform verwendet, was bedeutet, dass über das Internet Anwendungen angeboten werden, die einfach benutzt werden können. Es werden u. a. *Google*² und *BitTorrent*³ als Beispiele angeführt.

Die Nutzung kollektiver Intelligenz. Eine WEB2.0-Anwendung nutzt die kollektive Intelligenz seiner Benutzer. Beispiele sind wieder *Google* und *Amazon*. *Google* greift bei der Bewertung von Websites auf deren Eigenschaften zurück. *Amazon*⁴ bietet seinen Kunden die Möglichkeit, Erfahrungsberichte zu den angebotenen Produkten abzugeben. Es werden aber auch Websites wie *del.icio.us*⁵, *Flickr*⁶ oder *SourceForge.net*⁷ aufgeführt.

Das Vorhalten spezialisierter Daten. Das Datenbankmanagement gilt laut O'Reilly als eine der Kernkompetenzen der WEB2.0-Firmen. Viele Dienstleister besitzen hochspezialisierte Datenbanken, die mit einzigartigen Informationen gefüllt sind. Als Beispiel wird hier Amazon aufgeführt, die die Datenbank von ISBN-Nummern um weite-

¹s. O'Reilly (2005)

²Website: <http://www.google.com>

³Website: <http://www.bittorrent.com/>

⁴Website: <http://www.amazon.com/>

⁵Website: <http://del.icio.us>

⁶Website: <http://www.flickr.com>

⁷Website: <http://www.sourceforge.net>

re Merkmale, wie Coverbilder, Artikelbeschreibungen und Produktbewertungen durch Kunden erweitert hat. Durch diese Daten wird es potenziellen Mitbewerbern erschwert in diesen Markt einzutreten, da sie nicht einfach verfügbare Daten lizenzieren können, sondern auch erhebliche Zusatzinvestitionen tätigen müssen. Aber nicht nur das Vorhalten solcher Daten, sondern auch das Verknüpfen⁸ und Erweitern gilt als ein wichtiges Merkmal des WEB 2.0.

Abschaffung des Software-Lebenszyklus. Software wird nicht mehr als Paket, sondern als Dienst angeboten, der ständig weiterentwickelt wird. Es gibt keine Auslieferungen, die eine Reihe neuer Funktionen liefern. Stattdessen werden neue Funktionen in die Anwendung integriert, sobald ihre Entwicklung abgeschlossen ist. Im Gegenzug können Funktionalitäten auch wieder entfernt werden, falls die Nutzer sie nicht annehmen. Als Schlagwort ist in diesem Zusammenhang hier das *ewige Beta* zu nennen. Dabei ist es sehr wichtig, dass die betreffende Software und die dahinterstehenden Daten ständig gepflegt werden, um die Leistungsfähigkeit zu gewährleisten.

Lightweight Programming Models. Bei der Umsetzung von WEB2.0-Projekten sollte auf die Verwendung leichtgewichtiger Programmiermodelle geachtet werden. O'Reilly führt hier u.a. Amazon.com auf. Hier wird zum Einen eine umfangreiche SOAP- und zum Anderen eine leichtgewichtige REST-Schnittstelle angeboten, um anderen Anwendungen den Zugriff auf die Katalogdaten zu ermöglichen. O'Reilly (2005) bezieht sich auf Aussagen von Amazon, nach denen die REST-Schnittstelle von ca. 95% der Benutzer bevorzugt genutzt wird.

Die Verwendung solcher leichtgewichtigen Schnittstellen erleichtert laut O'Reilly, die Einbindung solcher Technologien in andere Websites.

Software über die Grenzen einzelner Geräte hinaus. Die entstehende Software soll geräteunabhängig sein, so dass die Software nicht mehr an einen Rechner oder eine bestimmte Hard- oder Softwareplattform gebunden wird. Es besteht hier ein freier Übergang zwischen mehreren Geräten.⁹

Benutzerführung (Rich User Experiences). Die Art der Benutzerführung ist eine der Schlüsselpunkte im WEB2.0. Durch den Einsatz von AJAX¹⁰ lassen sich Web-Anwendungen erstellen, die klassischen PC-Anwendungen sehr nahe kommen. Diese Möglichkeiten der Erstellung von Webapplikationen führt dazu, dass es inzwischen eine große Vielfalt von im Internet verfügbaren Anwendungen gibt, die direkt im Webbrowser ausgeführt werden.

⁸Das Erstellen sog. *Mashups*

⁹vgl. Transparenzkriterien in Tanenbaum und van Steen (2002)

¹⁰s. Kapitel 4.2

4.2 Technologien

Eine für diese Arbeit wichtige Technologie aus dem WEB2.0-Umfeld ist AJAX. Es kombiniert bereits von vielen Webbrowsern unterstützte Standards: Javascript und XML. Durch AJAX werden dem Entwickler Möglichkeiten der Benutzerführung geboten, die einer Website eher das Erscheinungsbild und das Verhalten einer klassischen Desktopanwendung verleihen.

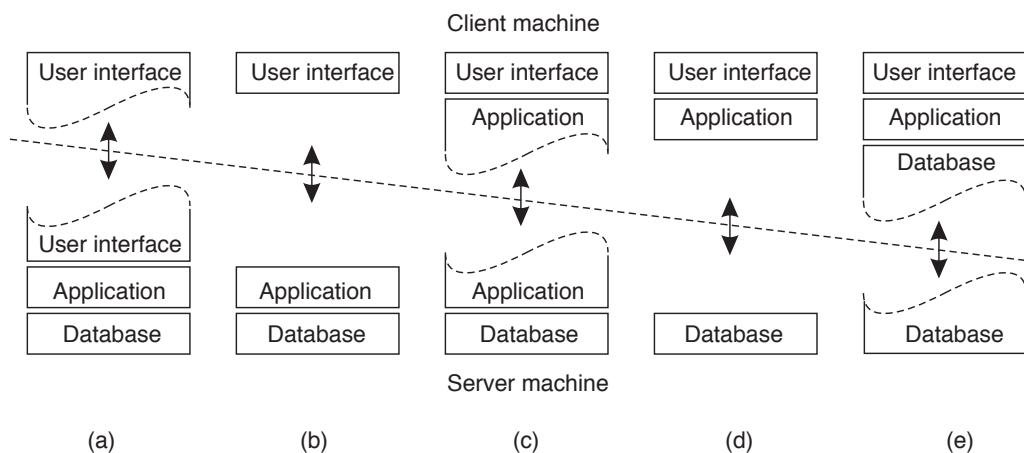


Abbildung 4.1: Alternative Client-Server-Architekturen

(Quelle: [Tanenbaum und van Steen \(2002\)](#), S .51)

In [Tanenbaum und van Steen \(2002\)](#) werden die Möglichkeiten des Aufbaus von Client-Server Anwendungen beschrieben. Die Abbildung 4.1 zeigt diese Arten. Durch die strikte Trennung von Präsentation und Anwendungslogik lassen sich klassische Webanwendungen meist im Teil (b) der Abbildung ansiedeln. Eine AJAX-Anwendung positioniert sich anders. Es ist hier auch möglich, die Trennung wie bei einer klassischen Web-Anwendung (Teil (b)) vorzunehmen. Es gibt aber noch zwei Alternativen. Es wäre eine Verschiebung in Richtung einer schlankeren Oberfläche (Teil (a)) möglich. In diesem Fall würden z.B. Reaktionen auf Oberflächenereignisse an den Server weitergeleitet und dort verarbeitet. Es wäre auch denkbar, dass Teile der Oberfläche vom Server dynamisch erzeugt und vom Client bei Bedarf nachgeladen werden (ebenfalls Teil (a)).

Eine weitere Möglichkeit wäre es, auf dem Client mehr Anwendungslogik (Teil (c)) zu implementieren. Dieses ist durch den umfangreichen Einsatz von Javascript möglich, so dass der Server von einfachen und häufigen Anfragen entlastet wird.

4.3 Beispiele

Es gibt bereits eine Vielzahl von AJAX-basierten Anwendungen im Internet. Im folgenden Abschnitt sollen einige Beispiele aufgeführt werden, welche als Grundlage für die weiteren Betrachtungen in dieser Arbeit dienen sollen.

4.3.1 Google Maps

Google bietet eine Reihe von Anwendungen basierend auf AJAX-Technologie an. Eine dieser Anwendungen ist *Google Maps*¹¹, welche hier stellvertretend für eine öffentlich zugängliche einfache¹² Web-Anwendung stehen soll. Es bietet eine Schnittstelle, über die ein Anwender eine Region oder einen Ort suchen kann. Dieser Ort wird dann, sofern er gefunden werden konnte, dem Benutzer auf einer Landkarte, einem Satellitenbild oder einer Kombination aus beidem, der so genannten Hybridansicht, angezeigt. Der Benutzer hat –ähnlich wie bei der vergleichbaren Desktopanwendung *Google Earth*– die Möglichkeit, den Kartenausschnitt durch Klicken und Ziehen mit der Maus zu bewegen. Es kann außerdem durch Drehen des Musrades der Zoomfaktor verändert werden und es besteht die Möglichkeit, sich Favoriten für häufig besuchte Orte anzulegen. Weitere Zusatzfunktionen sind ein Routenplaner und eine Unternehmenssuche. Die Benutzerschnittstelle von *Google Maps* ist in Abbildung 4.2 dargestellt.

Google Maps ist für einen einzelnen Benutzer an einem vertikal ausgerichteten Bildschirm ausgelegt. Es bietet keine speziellen Funktionalität für das gemeinsame Arbeiten an einem oder auch verschiedenen Rechnern oder Orten.

4.3.2 MindMapping (Virginio Carfagno)

Virginio Carfagno hat im Rahmen seiner Diplomarbeit ([Carfagno, 2007](#)) ein Mind-Mapping-Anwendung entwickelt. Der Benutzer kann sich mit Hilfe dieser Anwendung Mindmaps im Webbrowser erzeugen.

In einer Mindmap werden Begriffe auf einer Art Karte miteinander verknüpft. Die so entstehenden Verknüpfungen bilden i. d. R. einen Baum¹³ ab. Begonnen wird eine Mindmap, in dem in der Mitte des Arbeitsbereichs ein Startbegriff aufgeschrieben wird. Die Karte wird erweitert, in dem zu einem Begriff weitere assoziierbare Begriffe hinzugefügt werden.

¹¹Link: <http://maps.google.com>

¹²bezogen auf das Erscheinungsbild für den Benutzer

¹³weniger speziell: einen Graphen

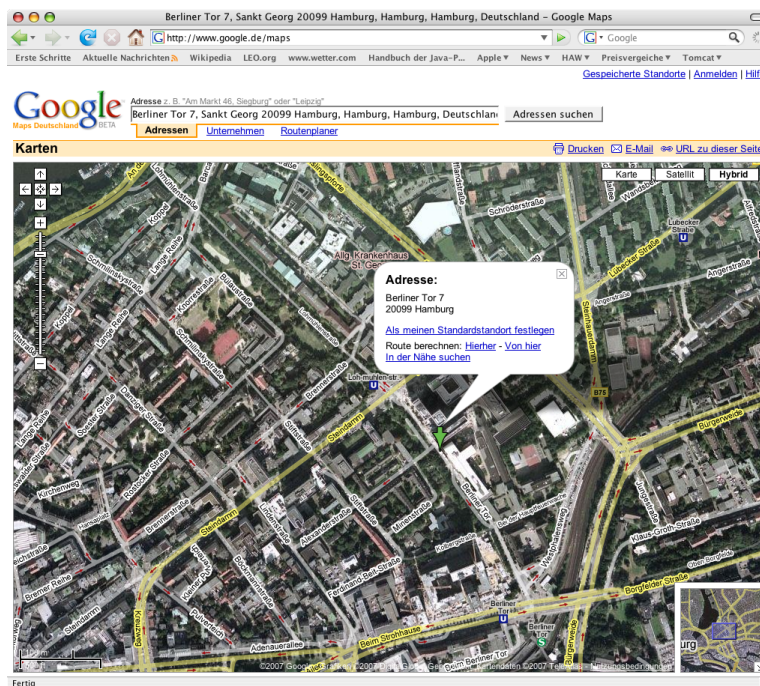


Abbildung 4.2: Screenshot: Google Maps

Der im Rahmen der Arbeit erstellte Prototyp bietet diese Möglichkeiten. Es ist hier zusätzlich noch möglich an einem Begriff Notizen oder Weblinks zu hinterlegen. Ein weiteres Feature ist die Möglichkeit, einzelne Äste der Mindmap zu verbergen.

Diese Anwendung bietet prinzipiell die Option des kooperierenden Arbeitens. Das gemeinsame erstellen einer Mindmap ist nicht unüblich und von der Arbeitsweise leicht realisierbar.

4.3.3 AJAX-basierte Office Anwendungen

Es existieren eine Reihe von frei benutzbaren AJAX-basierten Office-Anwendungen im Internet. Zwei Beispiele hierfür sind die Office-Pakete von Ajax13¹⁴ (*ajaxWrite*, *ajaxSketch*, *ajaxXLS*, *ajaxPresents*) und das von Google bereitgestellte *Google Docs*¹⁵ (*Google Write und Tables*), welches aus einer Textverarbeitung und einer Anwendung für Tabellenkalkulation besteht.

Beide Anwendungspakete versuchen, bestehende Office-Pakete, wie z.B. OpenOffice oder Microsoft Office nachzubilden. Sie können die Dateiformate dieser Pakete öffnen und speichern und bieten eine ähnliche Benutzeroberfläche.

¹⁴<http://www.ajax13.com>

¹⁵<http://docs.google.com>

Google Docs bietet zudem noch die Möglichkeit des gemeinsamen Bearbeitens eines Dokuments, indem ein Benutzer sein geöffnetes Dokument für weitere Benutzer freigibt. Es fehlen hierbei allerdings Möglichkeiten zur Kommunikation.

4.3.4 Meebo

Meebo¹⁶ ist eine Plattform, die eine Anbindung zu einer Reihe von Instant Messaging Protokollen bietet. Zum Zeitpunkt der Erstellung dieser Arbeit werden die Protokolle von AIM, Yahoo!, Google Talk, MSN, ICQ und Jabber unterstützt. Vergleichbare Desktopanwendungen wären Miranda, Gaim, oder Trillian.

Der größte Unterschied zwischen diesen klassischen Desktopanwendungen und Meebo ist wahrscheinlich, dass Meebo Gesprächsprotokolle serverseitig speichert. Es ist dem Benutzer somit möglich, diese an jedem beliebigen Ort einzusehen.

4.4 Zusammenfassung und Fazit

Es wurden eine Reihe von Anwendungen vorgestellt, die WEB2.0-Technologien einsetzen. Diese sind allerdings darauf ausgerichtet, einen Ersatz für bestehende Desktopanwendungen zu bieten und dabei die Vorteile der zentralen Verteilung zu nutzen. Die Benutzer müssen sich hierbei nicht mehr um die Installation von neuer oder um Updates bestehender Software kümmern. Sie erhalten stets die neueste Version.

Als weiterer Punkt sei hier die Geräteunabhängigkeit aufgeführt. Es ist für einen Benutzer der Webanwendungen unwichtig, auf welchem Rechner diese ausgeführt werden, sofern seine Daten zentral verfügbar sind, kann er an jedem beliebigen Ort weiterarbeiten.

Die Gemeinsamkeit aller vorgestellten Produkte ist ihre Konzeption als Einzelplatzanwendungen. Sie bieten nur in sehr begrenztem Umfang Möglichkeiten zur Zusammenarbeit an. Google Docs bietet Ansätze, die in diese Richtung zeigen. Der Fokus liegt hier jedoch eher auf dem entfernten Arbeiten. Zur Steigerung der Produktivität ist hier noch ein zusätzlicher Kommunikationskanal wie eine Sprach- oder Instant-Messaging-Verbindung notwendig.

Am ehesten geeignet für das gemeinsame Arbeiten mehrerer Personen an einem Ort wäre das MindMapping-Tool von Virginio Carfagno. Es bietet eine einfache Benutzerschnittstelle, die es mehreren Personen an einem großen Bildschirm erlauben kann, gemeinsam eine

¹⁶<http://www.meebo.com>

Mindmap zu erstellen. Es soll im weiteren Verlauf dieser Arbeit ein Beispielszenario aufgestellt werden, an dem sich die Eignung der AJAX-Technologie für das kooperierende Arbeiten am gleichen Ort und zur selben Zeit zeigt.

5 Beispielszenario: Virtueller Leuchttisch zur Erstellung einer Fotozeitschrift

Die Untersuchung der Einsetzbarkeit der AJAX-Technologie für das kooperierende Arbeiten soll in dieser Arbeit anhand eines Beispiels verdeutlicht werden. Bei diesem Szenario handelt es sich um eine fiktive Umgebung, in der das Seitenlayout für eine Fotozeitschrift erörtert werden kann.

Die Anwendung soll sich möglichst gut in den Arbeitsablauf des Unternehmens eingliedern. Es ist wichtig, dass sie auf die Wünsche und Bedürfnisse der Benutzer zugeschnitten wird, um von diesen auch angenommen zu werden.

In [Kahlbrandt \(2001\)](#) wird eine Trennung der Anforderungen an ein System in zwei Kategorien empfohlen:

- funktionale und
- nicht-funktionale Anforderungen

Es werden in diesem Kapitel zunächst die Rahmenbedingungen der Anwendung vorgestellt, um darauf aufbauend auf die funktionalen und nicht-funktionalen Anforderungen einzugehen. Es handelt sich dabei um eine unscharfe Beschreibung der Anforderungen. Die entsprechende Umsetzung der Anforderungen geschieht in Kapitel 6.

5.1 Randbedingungen

Die Anwendung ist von einer Reihe von Rahmenbedingungen umgeben. Diese sollen in diesem Abschnitt kurz vorgestellt werden.

5.1.1 Qualifikation der Benutzer

Der Benutzer des Systems ist ein Mitglied der Redaktion der Zeitschrift. Er ist mit der Arbeit am PC, insbesondere mit dem Umgang mit Grafik- und Layoutprogrammen, wie z. B. Adobe Photoshop, InDesign oder QuarkXpress, vertraut.

5.1.2 Verteilung der Anwendung

Es sollen mehrere Clients gleicher Art aufgebaut werden, damit ein unkomplizierter Zugang zu der neuen Anwendung gewährleistet ist. Die hierdurch entstehenden Wartungskosten sollen möglichst gering gehalten werden.

5.2 Funktionale Anforderungen

Im Folgenden sollen einige der identifizierten fachlichen Anforderungen an das System vorgestellt werden. Es handelt sich hierbei um die Grundfunktionalitäten der Anwendung.

5.2.1 Bereitstellen von Bild- und Textdaten

Um Inhalte in den Kontext der Anwendung bringen zu können, soll es möglich sein, über eine definierte Schnittstelle Text- und Bilddaten einpflegen zu können.

Diese Daten sollen dem jeweiligen Benutzer zuzuordnen sein und in einem ihm zugeordneten Bereich verwaltet werden können. Das Einpflegen der Daten soll möglichst über eine separate Anwendung geschehen. Diese wird z. B. von einem Notebook des Benutzers aus aufgerufen.

5.2.2 Zugriff auf hinterlegte Bild- und Textdaten

Der Zugriff auf die hinterlegten Daten soll über private Bereiche der jeweiligen Benutzer geschehen. Die Benutzer sollen hier auf zuvor festgelegte Sammlungen von Daten zugreifen können.

5.2.3 Einfügen von Bildern und Texten in den Layout-Bereich

Die im Personal Space des Benutzers angezeigten Daten sollen möglichst per Drag-and-Drop an eine gewünschte Position innerhalb des Group Space gezogen werden können. Es soll möglich sein, ein Element häufiger in eine Zeitschrift oder auch in eine Seite einzufügen.

5.2.4 Import und Export von Layoutdaten

Die erstellten Layouts sollen sich abspeichern und laden lassen, um den Benutzern die Möglichkeit zu geben, die Arbeit zu einem späteren Zeitpunkt und ggf. auch an einem anderen Ort fortzuführen.

Die Anwendung soll außerdem in der Lage sein, Dateien der in dem Unternehmen verwendeten Layout-Software zu importieren und zu exportieren. Das ist sinnvoll, da sich so Fragen zu einer bestehenden Arbeit einfach klären lassen. Die Ergebnisse aus solchen Besprechungen können dann, nach dem Export, direkt am Arbeitsplatz weiterbearbeitet werden.

5.2.5 Ändern der Orientierung des Layout-Bereichs

Die Anwendung soll als Eingabemedium einen Touchscreen benutzen. Dieser ist in einen Tisch eingelassen und wird von mindestens zwei sich gegenüber sitzenden Personen bedient. Dieser Aufbau hat zur Folge, dass es kein objektives *Oben* oder *Unten* gibt, sondern lediglich eine subjektive Orientierung.

Die Anwendung soll für jeden Benutzer gleichermaßen verwendbar sein. Um dieses Ziel zu erreichen, ist es erforderlich, dass sich das Bild an die Orientierung des jeweiligen Benutzers anpassen lässt. Das betrifft die Drehung aller angezeigten Bildelemente, wie z. B. Menüelemente, Bilder, Texte und deren Positionen auf dem Bildschirm.

In einer typischen Bearbeitungssituation mit zwei sich gegenüber sitzenden Bearbeitern würde beispielsweise die eine Person einige Inhalte in das Layout einfügen und der zweiten Person als Vorschlag präsentieren. Die zweite Person würde sich den Bildschirminhalt dann auf die für sie passende Orientierung *drehen*, um gegebenenfalls selbst Änderungen vorzunehmen¹.

¹Andeutungen auf dieses Verhalten finden sich in [Kruger u. a. \(2003\)](#)

Es soll hierbei möglich sein, dass die Arbeit einer Person am öffentlichen Bereich nicht die Arbeit der anderen Benutzer blockiert. Es ist daher notwendig, dass nur Teile des Bildschirm-inhalts gedreht werden können. Eine (simulierte) physische Drehung des gesamten Inhalts wäre daher nicht sinnvoll.

5.2.6 Transaktionsverwaltung

Eine weitere Anforderung an das System ist die Bereitstellung einer Transaktionsverwaltung. Die Benutzer sollen in der Lage sein, die Arbeit an der Anwendung in einzelne Verarbeitungsschritte aufzuteilen. Es soll dann möglich sein, auf den Stand einer vorhergegangenen Transaktion zurückzukehren. Diese Eigenschaft des Systems soll es erleichtern, dass die Benutzer mit dem Layout experimentieren können. Die Transaktionsverwaltung erleichtert das Wiederherstellen des Ursprungszustands, so dass die Benutzer bei Änderungen nicht befürchten müssen, eine gute Idee unwiederbringlich zu verwerfen.

5.3 Nicht-funktionale Anforderungen

Zusätzlich zu den bereits beschriebenen funktionalen Anforderungen gibt es eine Reihe von nicht funktionalen Anforderungen. Diese ergeben sich aus der besonderen Arbeitssituation an der Anwendung, sowie der angestrebten Verteilung der Anwendung.

5.3.1 Förderung der Kommunikation

Der Einsatz der Anwendung soll die aufgabenbezogene Kommunikation der Mitarbeiter untereinander fördern. Es soll somit schnell und effektiv zu verwertbaren Ergebnissen gelangt werden können.

Um dieses Ziel zu erreichen, ist die Umsetzung als Tabletop-Anwendung sinnvoll. Diese unterstützt, wie in Kapitel 3.3 beschrieben, am besten die natürliche Arbeitsweise, da sie zum Einen eine gewohnte Umgebung bietet und es zum Anderen den Benutzern den direkten Augenkontakt untereinander ermöglicht. Alle auf dem Tisch dargestellten Elemente sind für jeden Teilnehmer gut sichtbar und es ist allen Teilnehmern am Tisch möglich, die Arbeit der anderen zu beobachten. Es ist somit leichter, sich gegenseitig zu assistieren. Die in Kapitel 3.3.1 dargestellten Besonderheiten sind zu beachten.

5.3.2 Anforderungen an Eingabegeräte

Die Interaktion am Tisch soll nach Möglichkeit nur über den Touchscreen geschehen. Es soll hiermit ein Herumreichen der Tastatur (und ggf. der Maus) verhindert werden, da dieses wieder zu einer *moderierten* oder *rundenbasierten* Arbeitsweise führen könnte. In [Kruger u. a. \(2003\)](#) wurde ein solches Verhalten für die Arbeit am PC und in [Russell u. a. \(2002\)](#) für die Arbeitsweise am BlueBoard (vgl. Kapitel [3.1.1](#)) gezeigt.

Die Oberfläche soll daher einfach und intuitiv zu bedienen sein und den Einsatz zusätzlicher Eingabegeräte nicht erfordern.

5.3.3 Anforderungen an die Usability

Die Zufriedenheit der Nutzer mit der Anwendung ist ein wichtiges Anliegen. Sie soll daher nach den Ideen des *User Centered Design* geschehen. Die Norm [DIN EN ISO 13407 \(1999\)](#) definiert die Grundsätze der benutzerorientierten Gestaltung. Die Entwicklung der Anwendung soll sich nach diesen Grundsätzen richten.

6 Systementwurf

Die Anwendung ist in ihrer Ausrichtung eine AJAX-basierte Webanwendung. Die eine klassische Client-Server-Architektur bedingt. Das Ziel ist es allerdings nicht, diese öffentlich im Internet zur Verfügung zu stellen, sondern sie in einer speziellen und bekannten Hardwareumgebung auszuführen.

Dieses Kapitel beschäftigt sich mit dem Systementwurf der Anwendung. Es werden die wichtigen Designentscheidungen der Entwicklung aufgezeigt und das Grundkonzept der Anwendung verdeutlicht. Im Einzelnen werden zu diesem Zweck aus dem Szenario (Kapitel 5) Anwendungsfälle entwickelt, die in eine fachliche Architektur überführt werden. Darauf aufbauend wird dann ein Entwurf der technischen Architektur vorgestellt, bevor die Realisierung des Prototypen beschrieben wird.

6.1 Identifikation von Anwendungsfällen

Aus dem in Kapitel 5 vorgestellten Szenario lassen sich eine Reihe von Anwendungsfällen identifizieren. Diese lassen sich in *initiativeabhängige* und *initiativeunabhängige* Anwendungsfälle unterteilen. Ein Benutzer, der die *Initiative* hat, ist derjenige, zu dem der gemeinsame Bereich ausgerichtet ist. Abbildung 6.1 zeigt die identifizierten *initiativeabhängigen* Anwendungsfälle, dargestellt. Die Durchführung der dargestellten Aktionen ist abhängig vom Benutzerstatus und benötigt häufig die Interaktion beider Benutzer. Abbildung 6.2 zeigt die *initiativeunabhängigen* Anwendungsfälle. Die aufgelisteten Aktionen können unabhängig davon, ob der Benutzer die Initiative hat, ausgeführt werden.

Der für den Einsatz wichtigste Anwendungsfall, der *Wechsel der Initiative*, soll hier beispielhaft abgebildet werden. Die Bedeutung dieses Anwendungsfalls ist in der Zielumgebung begründet. Die Anwendung wird auf einem Tabletop-Bildschirm dargestellt. Es ist daher für die Verwendbarkeit der Anwendung unerlässlich, dass alle am Tisch arbeitenden Personen das zu bearbeitende Objekt genau betrachten können. Die Formulierung des Anwendungsfalls findet sich in der Tabelle 6.1.

Anwendungsfall	Wechsel der Initiative	
Zielbeschreibung	Es wird ein Wechsel der Initiative zwischen zwei Benutzern ausgeführt. Der Bildschirminhalt des gemeinsamen Arbeitsbereichs wird dem Benutzer zugewandt, der auch die Initiative bekommen hat.	
Vorbedingung	Ein Benutzer hat die Initiative und möchte diese abgeben. Ein zweiter Benutzer möchte die Initiative ergreifen.	
Erfolgsbedingung	Die Initiative wurde an den zweiten Benutzer abgegeben. Der Bildinhalt des gemeinsamen Bereichs wurde zu ihm ausgerichtet.	
Benutzerrollen	ein Benutzer (hat Initiative) und mindestens ein weiterer Benutzer	
Auslöser	Der Benutzer hat einen Arbeitsschritt im gemeinsamen Bereich getätigt und ein anderer möchte hierauf aufbauend weitere Arbeitsschritte durchführen.	
Beschreibung	Schritt	Tätigkeit
	1	Erster Benutzer hat Arbeitsschritt beendet
	2	Erster Benutzer signalisiert Bereitschaft zur Abgabe der Initiative
	3	Zweiter Benutzer nimmt die Initiative an
	4	Der aktuelle Systemzustand wird gesichert und die Transaktion damit abgeschlossen
	5	Der Bildausschnitt des gemeinsamen Bereichs ist zum zweiten Benutzer ausgerichtet

Tabelle 6.1: Anwendungsfall: Wechsel der Initiative

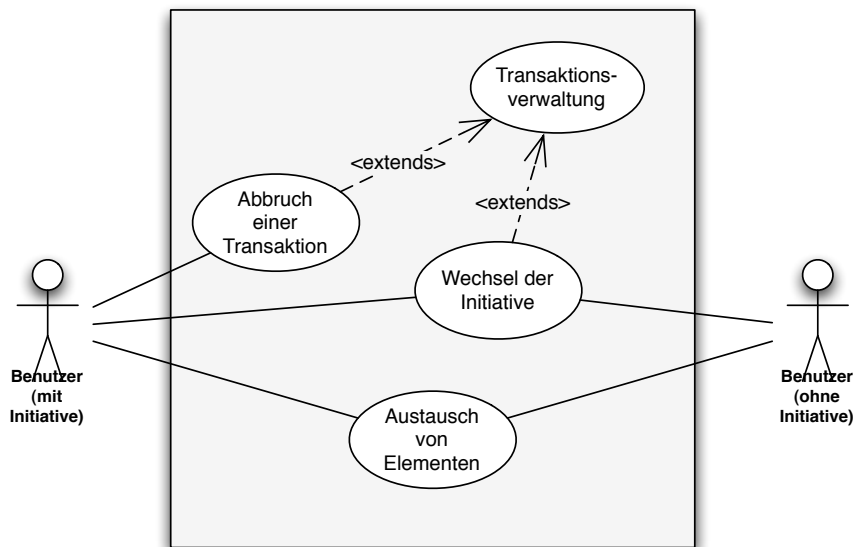


Abbildung 6.1: Use-Case-Diagramm: Initiativeabhängige Use Cases

Es soll im Rahmen der Implementierung noch überprüft werden, ob die Kombination aus Initiativwechsel und Abschluss der Transaktion sinnvoll ist. Es könnte sich hierbei herausstellen, dass die Benutzer einen Initiativwechsel durchführen, um anderen Benutzern Zwischenergebnisse zu erläutern. In diesem Fall könnte ein Abschluss der Transaktion nicht sinnvoll sein.

Die übrigen Anwendungsfälle aus den Abbildungen 6.1 und 6.2 sollen hier nicht weiter erläutert werden und sind daher im Anhang A aufgeführt.

6.2 Fachliche Architektur

In der Beschreibung der fachlichen Architektur wird die Anwendung in einzelne Module aufgeteilt. Es werden die einzelnen Bausteine, ihre Verbindungen und Schnittstellen dargestellt.

6.2.1 Aufteilung in Teilsysteme

Das Gesamtsystem lässt sich in drei Subsysteme unterteilen: Eine externe Clientanwendung, das Backend und die Tischanwendung aufteilen. Das Zusammenspiel der drei Anwendungsteile ist in der Abbildung 6.3 dargestellt.

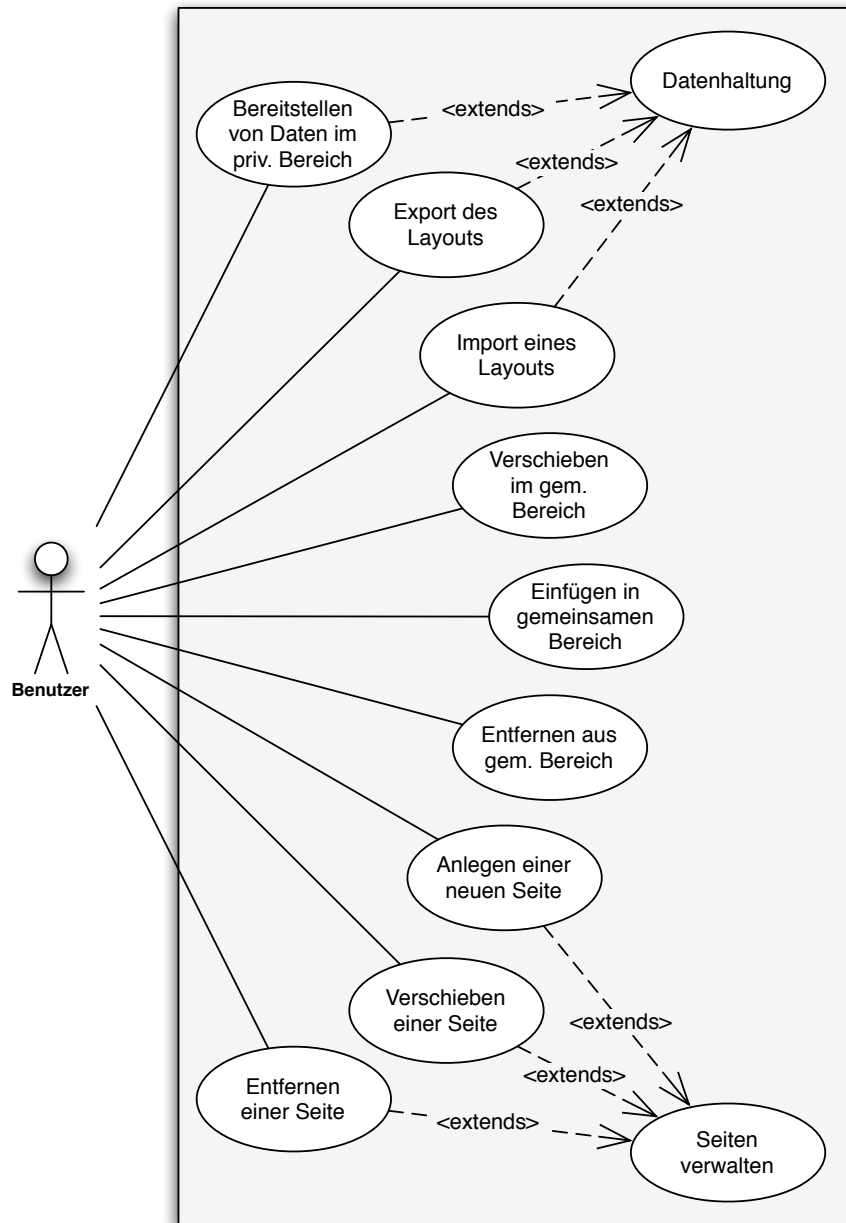


Abbildung 6.2: Use-Case-Diagramm: Initiativeunabhängige Use Cases

Die *private Anwendung* soll dem Benutzer die Möglichkeit bieten, seine eigenen Texte und Bilder vorzubereiten und einzupflegen, sie zu kategorisieren und für die Bearbeitung im öffentlichen Bereich freizugeben.

Die Komponente für die private Anwendung soll z.B. auf dem Notebook des Benutzers ausgeführt werden und somit unabhängig von der Tischanwendung sein.

Die *öffentliche Anwendung* bildet die Komponente, in der sich die bereitgestellten Vorlagen der Benutzer wiederfinden. Hier findet die eigentliche Erstellung der Zeitschrift statt. Auf dem Tisch werden die Seitenlayouts aus der Menge der freigegebenen Vorlagen zusammengestellt. Einzelne Seitenlayouts, sowie die Zeitschrift als Ganzes können gespeichert und geladen werden.

Das zentrale Element des Aufbaus ist das Backend, das Schnittstellen zum Up- und Download von Bild und Textdaten bieten soll. Diese Schnittstellen werden auf der einen Seite von den privaten externen Clients genutzt, um Daten in das System einzupflegen und für den Layoutvorgang vorzubereiten. Auf der anderen Seite werden diese hinterlegten Daten der *Tischanwendung* zur Verfügung gestellt, wo sie zu einer Zeitschrift zusammengefügt werden.

Über die genannten Schnittstellen werden auch die Layoutbeschreibungen geladen und gespeichert, so dass diese später weiterverarbeitet werden können.

6.2.2 Oberflächenkonzept

Die Oberfläche hat einen elementaren Bestandteil am Erfolg der Anwendung. Sie ist das, was der Benutzer von der Anwendung wahrnimmt. In diesem Abschnitt soll ein Konzept für die Oberfläche des Redaktionssystems vorgestellt werden.

Besonderheiten

Die Oberfläche soll unter Berücksichtigung der in Abschnitt 3.3.1 aufgeführten Besonderheiten gestaltet werden. Sie soll zunächst als *Pairware* erstellt werden, also eine *Groupware* für zwei Personen¹. Es soll jedoch möglich sein, in späteren Versionen eine Erweiterung auf mehrere Benutzer durchzuführen.

¹s. [Ishii und Kobayashi \(1992\)](#)

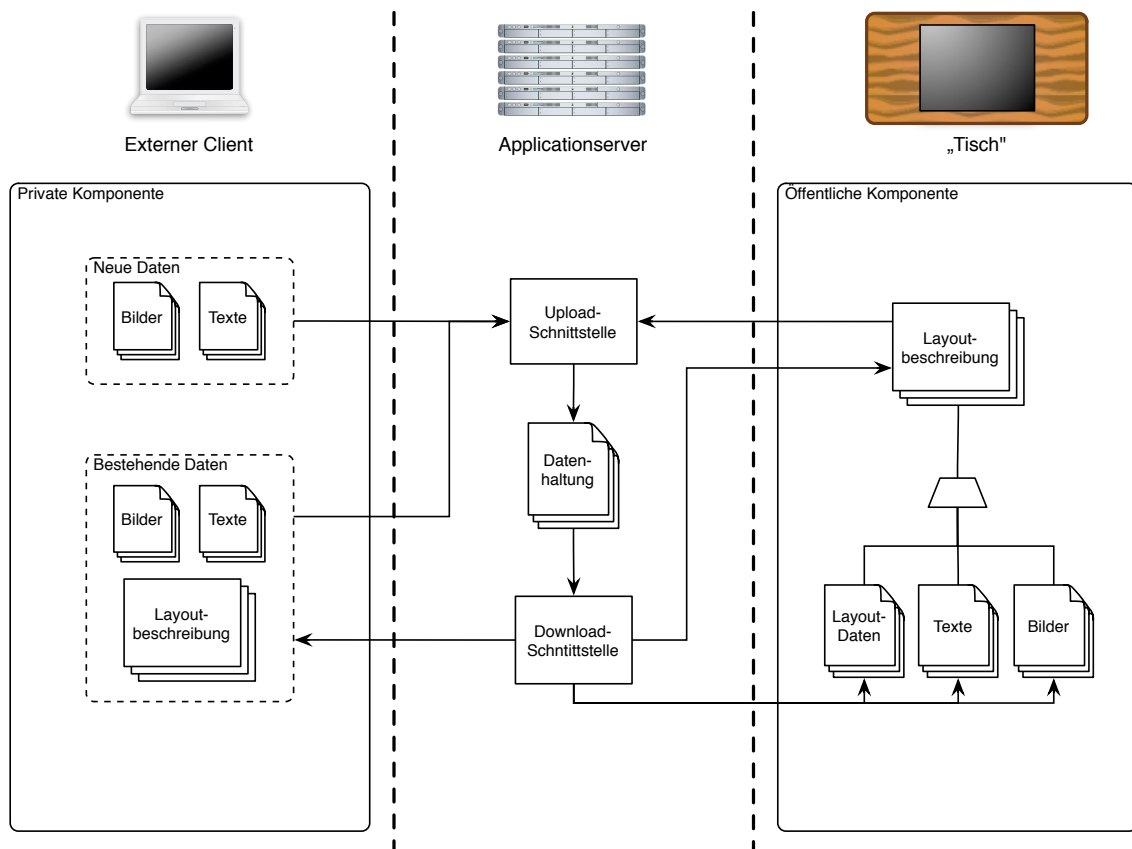


Abbildung 6.3: Überblick der Teilsysteme

Entwurf

Der Entwurf der Anwendung sieht eine Einteilung in verschiedene Bereiche vor. In der Mitte des Fensters befindet sich der gemeinsame Bereich, in dem das Layout der Zeitschrift entwickelt wird. An den langen Seiten befinden sich die privaten Elemente der Benutzer, die in das Layout eingefügt werden können. An den kurzen Seiten des Bildschirms sind Werkzeuge untergebracht, die die Steuerung der Anwendung erlauben². Aus der Anordnung des Layouts, mit den persönlichen Bereichen an den langen Seiten des Bildschirms, ergibt sich, dass sich die beiden Benutzer gegenüber sitzen. In späteren Versionen können auch an den kurzen Seiten persönliche Bereiche eingerichtet werden, so dass sich die Anwendung von insgesamt vier Personen bedienen ließe.

Bei der Entwicklung dieses Entwurfs wurde lediglich eine Rotation um 180° vorgesehen. Die Erweiterung würde Rotationen in 90°-Schritten zulassen. Andere Rotationswinkel, die keine Teiler von 90° sind, würden dazu führen, dass die Bilddaten interpoliert werden müssten. Abbildung 6.4 zeigt einen Entwurf der Oberfläche. Es ist in der Mitte der gemeinsame Bereich dargestellt. An den Rändern findet sich jeweils Platz für Werkzeuge und Materialien.

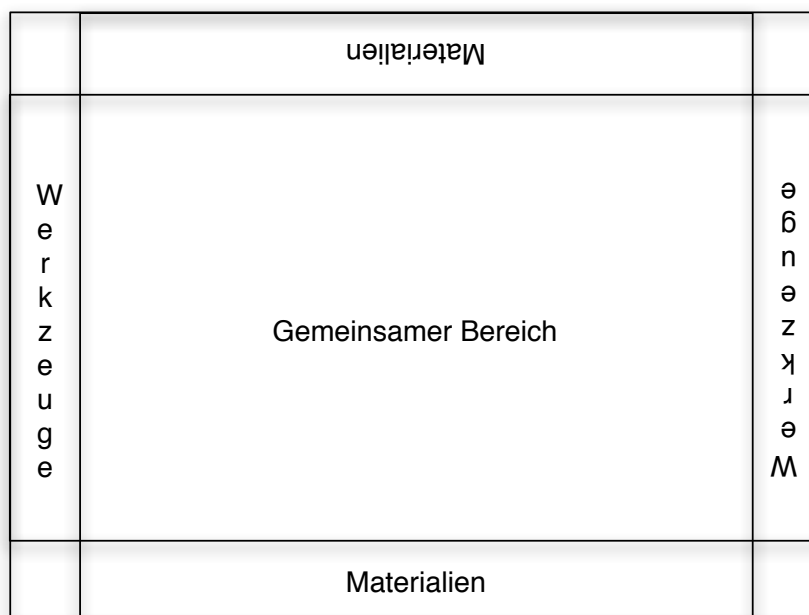


Abbildung 6.4: Oberflächenkonzept

²z.B. Wechsel der Initiative, Transaktionsabbruch, Import, Export, etc.

6.3 Technische Architektur

Die Beschreibung der technischen Architektur liefert Aufschluss über die technische Umsetzung der Architektur. Es werden Besonderheiten der Laufzeitumgebung dargelegt. Bei der Beschreibung der Komponenten soll nur die Tischkomponente näher beschrieben werden, da diese auch den Fokus des im Kapitel 6.4 beschriebenen Prototypen hat.

Es fallen daher die Teile *externer Client* und *Datenhaltung* aus der Beschreibung heraus.

6.3.1 Rahmenbedingungen

Dieses Kapitel soll einen Überblick der Umgebung und der Anforderungen liefern. Es wird dabei insbesondere auf die spezielle Hard- und Software-Situation eingegangen und alle wichtigen Hard- und Softwarekomponenten kurz vorgestellt.

Hardware

Die Hardware des Konferenzraumes umfasst eine Reihe von Blade-Servern, die u. a. einen Webserver bereitstellen können.

Die Eingaben sollen hauptsächlich über einen in einen Tisch eingelassenen Touchscreen geschehen. Der in Kapitel 2.1.3 dargestellte *InteracTable* ist der Zielumgebung sehr nahe. An diesem Tisch können zwei sich gegenüber sitzende Personen gemeinsam arbeiten.

Für die Bereitstellung der Daten werden außerdem noch externe Clients benötigt. Es handelt sich hierbei entweder um die PCs der Mitarbeiter oder um Notebooks der Gruppenteilnehmer. Diese externen Clients werden spontan in das System eingebunden und auch wieder entfernt. Sie sind daher nicht permanent mit diesem verbunden.

Software

Da es sich bei der gewählten Anwendung um eine AJAX-Anwendung handeln soll, wird auf der Client-Seite ein geeigneter Javascript-fähiger Web-Browser benötigt. Da die Interpretation und der unterstützte Sprachumfang von Browser zu Browser variieren kann, soll sich der Prototyp der Anwendung auf aktuelle Gecko-basierte Webbrowser³ beschränken. Diese sind auf einer Vielzahl von Betriebssystem- und Hardwareplattformen verfügbar und gewährleisten somit Unabhängigkeit von diesen Faktoren. Zusätzlich ist die Gecko-Engine öffentlich im

³z. B. Mozilla Firefox

Quelltext verfügbar. Es ist somit möglich, diese als virtuelle Maschine verwendete Software in ein speziell angepasstes Rahmenwerk zu integrieren. Ein weiterer Vorteil der Gecko-Engine ist die direkte Unterstützung des SVG-Formats⁴.

Auf der Server-Seite werden ein Webserver und eine Datenbank eingesetzt werden.

6.3.2 Einteilung in Schichten

Die Architektur der Anwendung wird als eine klassische Drei-Schichten-Architektur geplant. Es wird hierbei großer Wert auf die Wiederverwendbarkeit der einzelnen Komponenten gelegt. Dieses hat den Vorteil, dass Änderungen an einzelnen Modulen nur sehr geringe oder im besten Fall keine Auswirkungen auf das Gesamtsystem haben, was zu einer besseren Wartbarkeit führt. Zudem bietet dieses Vorgehen den Vorteil, dass einzelne Komponenten unter Beibehaltung der Schnittstellen ausgetauscht werden können. Die Abbildung 6.5 zeigt den Aufbau der Anwendung und die Einordnung der Komponenten in die entsprechenden Schichten. Da es sich bei der Applikation um eine WEB2.0-Anwendung handelt, bietet es sich an, einfache Abläufe der Geschäftslogik bereits im Browser ablaufen zu lassen. Das bietet den Vorteil, dass der Webserver weniger belastet wird und eine direkte Reaktion auf die getätigten Eingaben des Benutzers möglich ist.

Die Kommunikation der Oberfläche mit der Geschäftslogik soll über asynchrone Aufrufe⁵ geschehen. Diese haben jedoch keinen direkten Zugriff auf die Datenhaltung der Anwendung, sondern kommunizieren ebenfalls mit der *Kerngeschäftslogik*. In der Geschäftslogik sollen insbesondere das Laden und Speichern der Daten sowie eine Benutzerauthentifizierung enthalten sein.

6.3.3 Kommunikationsprotokoll

Die Kommunikation zwischen den einzelnen Teilsystemen wird über das HTTP-Protokoll geschehen. Es ist das Standardprotokoll für die Übermittlung von HTML-Seiten und wird von der verwendeten Browseranwendung direkt unterstützt.

⁴s. Kapitel 6.4.2

⁵z.B. mittels AJAX

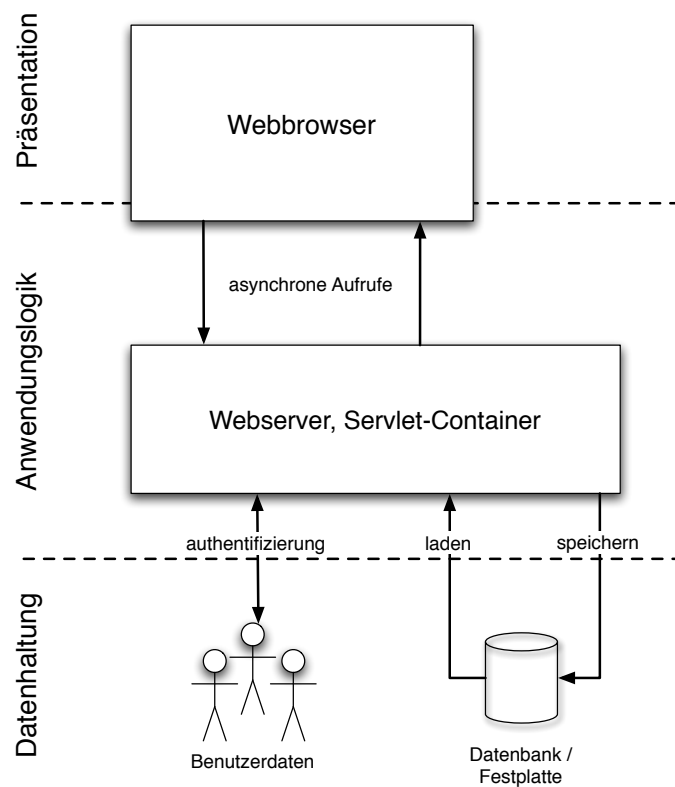


Abbildung 6.5: Einteilung der Anwendung in Schichten

6.3.4 Anwendungsdesign

Das Anwendungsdesign spiegelt die Kernlogik der Anwendung wieder. Es wird an dieser Stelle beispielhaft die Zusammensetzung der Zeitschrift aus ihren einzelnen Bestandteilen dargestellt.

Aufbau der Zeitschrift

Die Fotozeitschrift ist das zentrale Element der Anwendung. Sie ist der Container, in den die Arbeit der Anwender hineingegeben wird.

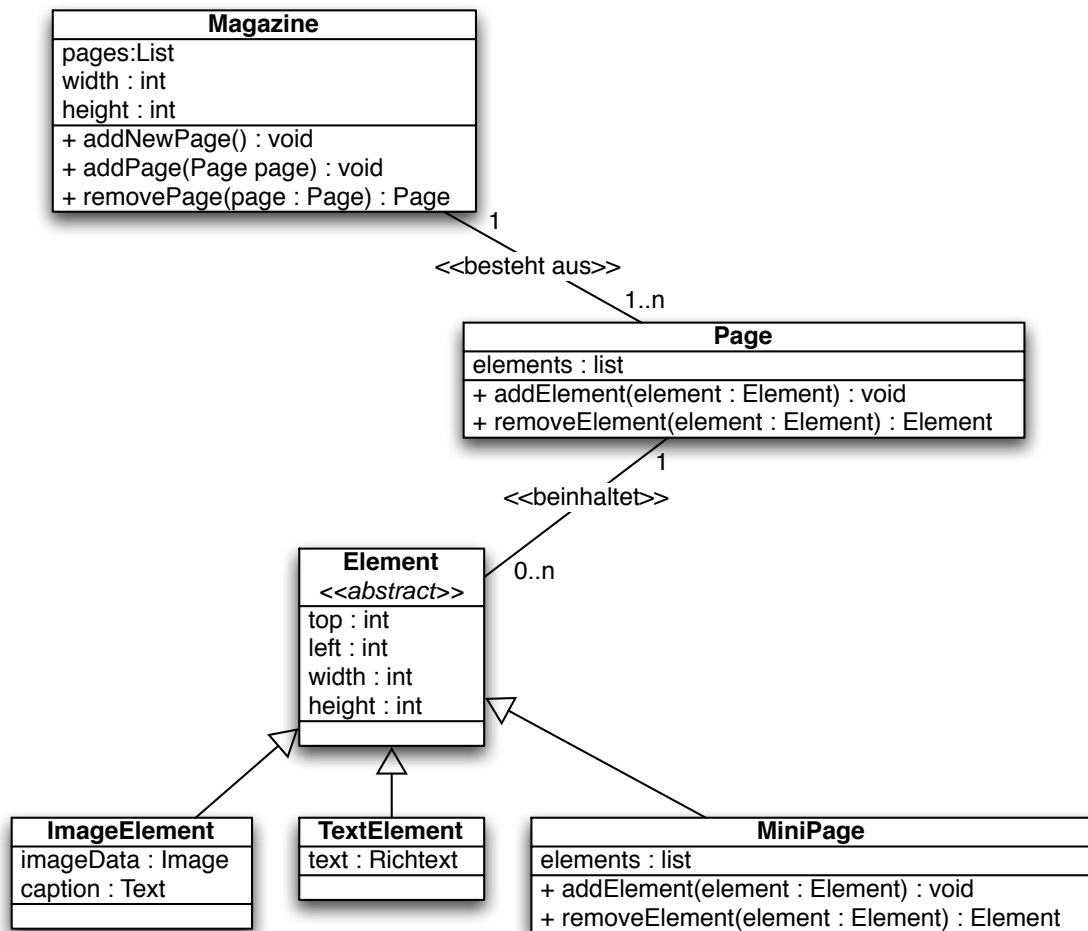


Abbildung 6.6: Klassendiagramm Magazin

Die Abbildung 6.6 zeigt den Aufbau der Fotozeitschrift. Die Klasse *Magazine* bildet die Wurzel. Sie definiert die Maße der Zeitschrift und dient als Container für eine Liste von *Page*-Objekten.

Die Klasse *Page* repräsentiert eine Seite der Zeitschrift. Sie ist wiederum ein Container für Objekte vom Typ *Element*.

Ein *Element* stellt die kleinste in einer Seite positionierbare Einheit dar. Es werden hier beispielhaft drei verschiedene Implementationen von *Element* dargestellt:

- *ImageElement* repräsentiert ein Bild
- *TextElement* repräsentiert einen Text
- *MiniPage* repräsentiert eine beliebige Gruppierung von Elementen

Ein Element kennt jeweils seine Position und seine Größe in der Seite.

In diesem Beispiel finden sich sowohl eine Vererbungshierarchie, als auch das Composite-Pattern ([Gamma u. a., 1995](#)) wieder.

6.4 Realisierung eines Prototypen

Um eine Entscheidungsgrundlage für die Einsetzbarkeit der gewählten Technologien treffen zu können, ist die Entwicklung eines experimentellen Prototypen sinnvoll. Es lassen sich so eventuelle Schwachstellen der Technologie bereits in einer frühen Phase der Entwicklung erkennen und das Entwicklersteam hat die Möglichkeit frühzeitig nach Lösungen oder Alternativen zu suchen.

In diesem Kapitel wird aufgezeigt, wie ein Prototyp aufzeigen kann, ob sich die in den Abschnitten 6.3 und 6.2 beschriebene Architektur für die Entwicklung eines Rich-Web-Client eignet.

6.4.1 Zielsetzung

Der aus den Anforderungen hervorgehende Prototyp soll zeigen, ob es machbar und sinnvoll ist, eine AJAX-basierte Webanwendung für eine Tabletop-Umgebung zu verwenden.

Es ist von existierenden klassischen Webanwendungen bekannt, dass eine umfangreiche und auch vergleichbare Datenverarbeitung auf dem Server möglich ist. Es soll sich bei dem Prototypen daher nur um eine reine Oberflächenstudie handeln, in der die Einsetzbarkeit der Technologie gezeigt werden kann.

Das Hauptaugenmerk liegt dabei auf der Wartbarkeit der GUI-Komponenten.

6.4.2 Herausforderungen

Es gibt in der Beschreibung der Anwendung eine Reihe von Herausforderungen, die in einer einfachen Desktopumgebung nur selten oder auch gar nicht auftreten. Ziel ist es mit Hilfe der hier aufgeführten Punkte die Basisanforderungen für den Prototypen zu ermitteln.

Datei-Upload

Erste Schwierigkeiten ergeben sich aus den Sicherheitsbeschränkungen des Web-Browsers. Es ist hier beispielsweise nicht möglich, ein Verzeichnis mit Bilddaten auszuwählen und auf den Server zu laden. Die für diesen Zweck zur Verfügung stehende HTML-Komponente erlaubt es nur, dass einzelne Dateien ausgewählt werden, jedoch keine Verzeichnisse oder Dateimengen.

Da dieses Vorgehen in einer datenzentrierten Anwendung wenig effizient ist, musste hierfür eine Lösung gefunden werden.

Ein Ansatz ist z.B. das Heraustrennen der Upload-Komponente aus der eigentlichen Anwendung. Es wäre somit möglich, den Upload über einen FTP-Client oder eine eigene kleine Upload-Anwendung⁶ durchzuführen. Eine weitere Möglichkeit ist, die Bilddateien als Anhang an eine spezielle E-Mail-Adresse des Benutzers zu senden. Diese Alternative bietet den Vorteil, dass die meisten Benutzer mit dem Umgang mit E-Mails vertraut sind. Es wäre somit möglich, die Bilder von jedem beliebigen Ort aus zur Verfügung zu stellen, auch wenn man sich nicht im entsprechenden Netzwerk befindet. Eine dritte Option wäre die Verwendung eines Java-Applets, das Zugriff auf das lokale Dateisystem hat, sofern es entsprechend signiert wurde.

Diese Vorgehensweisen werden auch von Datenintensiven Plattformen im Internet angewandt. So können z.B. Benutzer des Dienstes Flickr⁷ Bilder u.a. an eine bestimmte E-Mailadresse schicken, um diese zu ihrer Sammlung hinzuzufügen.

⁶realisiert z.B. in Java

⁷Website: <http://www.flickr.com>

Rotation von Objekten

Auf einem Bildschirm, der in einen Tisch eingelassen ist, gibt es kein objektives *Oben* oder *Unten*. Diese Orientierung des Bildes hängt subjektiv von der Position jedes Betrachters am Tisch ab. Es ist daher wichtig, dass jeder Betrachter die dargestellten Inhalte in eine für ihn günstige Position ausrichten kann.

Die Spezifikationen von HTML⁸ und CSS⁹ bieten leider keine Möglichkeit, einfachen Text rotiert auszugeben. Eine mögliche Lösung für dieses Problem wäre z. B. die generelle Verwendung von Grafiken. Es bietet sich hier die Verwendung von Vektorgrafiken auf der Basis von SVG an. SVG wird vollständig in XML beschrieben und lässt sich somit einfach generieren und sogar Client-Seitig manipulieren.

Besonderheiten der Oberfläche

Die Gestaltung der Oberfläche einer Anwendung, deren hauptsächliches Eingabegerät ein Touchscreen ist und die somit zu großen Teilen ohne Tastatur oder Maus zu bedienen sein soll, ist eine spezielle Herausforderung. Die Umsetzung einer solchen Anwendung als Web-Anwendung stellt eine zusätzliche Schwierigkeit dar, da die Erstellung von klassischen Web-Sites generell auf die Verwendung von Tastatur und Maus ausgelegt ist.

Es kann in der Anwendung auf die wenigsten Standardkomponenten zurückgegriffen werden. Es sollte daher u. a. untersucht werden, ob die Bedienung von Scrollbalken möglich ist, oder ob diese aufgrund ihrer mangelnden Größe durch grafische Komponenten ersetzt werden sollten.

6.4.3 Implementation

Es sollen für die Implementation des Prototypen nur die kritischen Teile der Herausforderungen herausgegriffen werden. Es soll der Aufwand der Untersuchung gering gehalten werden. Es soll somit auch vermieden werden, Fragestellungen zu erörtern, deren Antwort bereits bekannt ist.

⁸s. [Raggett u. a. \(1999\)](#)

⁹s. [Bos u. a. \(1998\)](#)

Auswahl eines geeigneten Frameworks

Es gibt eine Reihe von AJAX-Frameworks. Die meisten haben jedoch einen Punkt gemeinsam: es ist erforderlich, dass sie in JavaScript angesprochen werden. Auf Grund dieser Tatsache ergeben sich eine ganze Reihe von Schwierigkeiten, die bei der Entwicklung mit diesen Frameworks auftreten. Es ist z. B. nicht möglich, eine saubere Modularisierung des Codes vorzunehmen, wodurch die wiederverwendbarkeit von Komponenten erschwert wird. Es wird außerdem das Testen und Debuggen der Anwendungen erschwert.

Das Google Webtoolkit soll diese Problematik entschärfen. Es bietet Java-Entwicklern die Möglichkeit, Rich-Web-Clients direkt in Java zu realisieren. Der Java-Code wird dann von einem Crosscompiler in JavaScript übersetzt. Das Google Webtoolkit erzeugt, laut Google, JavaScript, das auf allen gängigen Webbrowsern lauffähig ist. Der zweite große Vorteil des Frameworks ist, dass sich alle bekannten Vorteile der Java-Entwicklung weiter nutzen lassen. Es wird also eine Organisation der Anwendung in Klassen und Pakete möglich, die in reinem JavaScript auf diese Weise nicht abbildbar wäre. Es bietet zudem Möglichkeiten die Anwendungslogik direkt im Java-Code zu debuggen und bietet eine einfache Integration von automatisierten Tests. Eine umfassende Untersuchung des Google Webtoolkit findet sich in [Carfagno \(2007\)](#).

Schwerpunkt der Implementation

Der Schwerpunkt der Implementation liegt auf der Organisation des Codes. Es soll festgestellt werden, ob es möglich ist wartbaren Code zu erzeugen. In diesem Zusammenhang sollte auch überprüft werden, in wie fern sich die [Gamma u. a. \(1995\)](#) vorgestellten Entwurfsmuster umsetzen lassen. Für die Umsetzung des Prototypen wird zunächst nur die Anzeige des Arbeitsbereichs mit entsprechenden, den Benutzer zugewandten Bildern angestrebt.

Auf die Implementierung einer umfangreichen Backend-Komponente wird bei diesem Prototypen verzichtet, da hier zu großen Teilen die bereits bekannten Methoden der Umsetzung möglich sind. Es werden in diesem Bereich somit keine Veränderungen gegenüber klassischen Webanwendungen erwartet.

7 Beurteilung und Fazit

7.1 Zusammenfassung

In der vorliegenden Arbeit wurde eine Grundlage geschaffen, auf der sich entscheiden lässt, ob sich der Einsatz von AJAX-basierten Lösungen für das kooperierende Arbeiten eignet. Hierzu wurden zunächst in Kapitel 2 und 3 Infrastrukturen und Einsatzkontexte für Collaborative Workspaces vorgestellt. Anschließend wurden in Kapitel 4 die Hintergründe des WEB 2.0 und der zugehörigen AJAX-Technologien erläutert. Es wurde mit diesen Themen im Hintergrund ein Beispielszenario, die Layoutgestaltung einer Fotozeitschrift auf einem virtuellen Leuchttisch, aufgestellt (Kapitel 5). Aus dem Szenario wurde dann in Kapitel 6 zunächst ein Systementwurf mit einer weitreichenden Vision der Architektur erstellt. Aus diesem Entwurf wurden dann in Abschnitt 6.4 die entscheidenden Punkte identifiziert und in einen Prototypen umgesetzt.

Die Entscheidung über die Einsetzbarkeit der AJAX-Technologie war weniger eine Entscheidung im Backend, da dort meist auf etablierte und bekannte Softwarekomponenten zurückgegriffen werden kann. Die Entscheidung für oder gegen diese Technologie wird daher im Frontend getroffen. Etablierte AJAX-Anwendungen haben bereits gezeigt, dass es möglich ist komplexe Anwendungen auf AJAX-Basis zu erstellen. Der Prototyp sollte daher vor allem Antworten zu Fragen der Wartbarkeit und den speziellen Anforderungen einer Tabletop-Anwendung finden. Die Entwicklung hat gezeigt, dass es möglich ist, mit Hilfe des Google Webtoolkits die Lücke zwischen Anwendungsdesign und Realisierung zu verkleinern. Es erlaubt, die Realisierung in Java. Es lassen sich somit viele Vorteile der Sprachumgebung nutzen. Als Beispiele sein hier die Möglichkeit des automatisierten Testens oder auch die Wiederverwendbarkeit von Komponenten genannt. Der Prototyp hat gezeigt, dass es sinnvoll ist, die Technologie weiter zu betrachten. Der Prototyp hat gezeigt, dass es grundsätzlich sinnvoll ist die vorgestellten Technologien zu verbinden. Darauf aufbauend sollen im Ausblick Wege aufgezeigt werden, die für einer Weiterführung der Arbeit sinnvoll sind.

7.2 Ausblick

Sie gibt jedoch einen guten Ausblick, auf die Potenziale der AJAX-Technologien in diesem Umfeld.

In weiteren Untersuchungen wäre es zunächst sinnvoll, die aufgezeigte Architekturbeschreibung zu vervollständigen, damit diese umgesetzt werden kann. Bei der Umsetzung ist die Durchführung umfangreicher Usability-Tests sinnvoll, um somit frühzeitig Schwierigkeiten bei der Akzeptanz der Benutzer zu stoßen.

Zusätzlich sollte überprüft werden, in wie fern sich der verwendete Webbrowser in einer Multitouch-fähigen Umgebung verhält. Diese Technologie macht ein flüssiges gemeinsames Arbeiten erst sinnvoll möglich, da sie gleichzeitige Benutzereingaben erlaubt (Scott u. a., 2003). In diesem Zusammenhang sollte auch geprüft werden, in wie fern die Entwicklung einer eigenen Gecko-basierten Virtuellen Maschine sinnvoll ist. Eine Anpassung in dieser Richtung kann sinnvoll sein, um ein besser angepasstes und integriertes Umfeld für den Benutzer zu schaffen, das nicht den Eindruck erweckt, als liefe es in einem Webbrowser.

Abschließend seien hier auch noch zwei weitere Möglichkeiten der Oberflächenbeschreibung erwähnt: XUL¹ vom Mozilla-Projekt und XAML² von Microsoft. Beides sind XML Beschreibungssprachen für Oberflächen, die sich auch für die Verwendung in Websites eignen. Diese Sprachen bieten weitaus freiere Beschreibungsmöglichkeiten für Oberflächen, als der HTML-Standard, bringen jedoch auch eine Reihe einschränkender Bedingungen mit sich.

Es wurden in dieser Arbeit nur geringfügig Überlegungen zur Benutzerauthentifizierung und zur Sicherheit des Systems gemacht. Diese sollten in späteren Versionen verstärkt berücksichtigt werden.

¹XML User Interface Language

²eXtensible Application Markup Language

Literaturverzeichnis

- [Agarawala 2006] AGARAWALA, Anand: *Enriching the Desktop Metaphor with Physics, Piles and the Pen*, University of Toronto, Masters Thesis, 2006. – URL <http://honeybrown.ca/Pubs/Thesis-BumpTop.pdf>. – Zugriffsdatum: 19.02.2007
- [Bartnik 2006] BARTNIK, Roman: *Weiterentwicklung einer Technologiebasis für interaktive Gruppenarbeitsräume*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/bartnik.pdf>. – Zugriffsdatum: 22.02.2007
- [Bos u. a. 1998] BOS, Bert ; LIE, Håkon W. ; LILLEY, Chris ; JACOBS, Ian: *Cascading Style Sheets, level 2; CSS2 Specification*. 12. Mai 1998. – URL <http://www.w3.org/TR/REC-CSS2/>. – Zugriffsdatum: 27.02.2007
- [Burfeindt 2006] BURFEINDT, Lars: *Konstruktion einer Middleware für computergestützte Gruppenarbeit in ubiquitärer Systemumgebung*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2006. – URL <https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/burfeindt.pdf>. – Zugriffsdatum: 22.02.2007
- [Carfagno 2007] CARFAGNO, Virginio: *Evaluation des Google Web Toolkits durch Entwicklung einer ajaxbasierten Mind-Mapping-Anwendung*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2007. – in Vorbereitung
- [DIN EN ISO 13407 1999] CEN (Hrsg.): *Benutzer-orientierte Gestaltung interaktiver Systeme (ISO 13407 : 1999)*. CEN Europäisches Komitee für Normung, rue de Stassart 36, B-1050 Brüssel, November 1999
- [Durwen 2001] DURWEN, Karl-Josef: *Im Spiegel der Möglichkeiten*. Ostfildern vor Stuttgart : edition tertium, 2001. – ISBN 3-930717-68-9
- [Fraunhofer-IPSI 2005] FRAUNHOFER-IPSI: *Fraunhofer InteracTable*. 2005. – URL <http://www.ipsi.fraunhofer.de/ambiente/english/projekte/projekte/ineractable.html>. – Zugriffsdatum: 17.12.2006
- [Gamma u. a. 1995] GAMMA, E. ; HELM, R. ; JOHNSON, R. ; VLISSIDES, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995

- [Hinrichs 2005] HINRICHS, Uta: *Interface Currents: Supporting Co-Located Collaborative Work on Tabletop Displays*, Department of Computer Science, University of Calgary, Technical Report, 2005. – URL http://www.utahinrichs.de/work/technicalReport/technicalReport_new.pdf. – Zugriffsdatum: 16.02.2007
- [Hinrichs 2006] HINRICHS, Uta: *Interface Currents: Evaluating a Fluid Interface for Tabletop Collaboration*, Otto-von-Guericke-University Magdeburg, Diplomarbeit, 2006. – URL http://www.utahinrichs.de/work/thesis/diplom_final.pdf. – Zugriffsdatum: 16.02.2007
- [Hinrichs u.a. 2005a] HINRICHS, Uta ; CARPENDALE, Sheelagh ; SCOTT, Stacey D.: *Interface Currents: Supporting Fluent Face-to-Face Collaboration*. Berlin Heidelberg : Springer-Verlag, August 2005. – URL http://www.utahinrichs.de/work/smartGraphics/sg_paper_final.pdf. – Zugriffsdatum: 18.02.2007
- [Hinrichs u. a. 2006] HINRICHS, Uta ; CARPENDALE, Sheelagh ; SCOTT, Stacey D.: *Evaluating the Effects of Fluid Interface Components on Tabletop Collaboration*. In: *International Working Conference on Advanced Visual Interfaces*. Venezia, Italy, Mai 2006. – URL <http://www.utahinrichs.de/work/avi2006/103-Hinrichs.pdf>. – Zugriffsdatum: 18.02.2007
- [Hinrichs u. a. 2005b] HINRICHS, Uta ; CARPENDALE, Sheelagh ; SCOTT, Stacey D. ; PAT-TISON, Eric: *Interface Currents: Supporting Fluent Collaboration on Tabletop Displays*. In: *5th Symposium on Smart Graphics*. Berlin Heidelberg : Springer-Verlag, August 2005, S. 185–197. – URL http://www.utahinrichs.de/work/smartGraphics/sg_paper_final.pdf. – Zugriffsdatum: 18.02.2007
- [Ishii und Kobayashi 1992] ISHII, Hiroshi ; KOBAYASHI, Minoru: *ClearBoard: a seamless medium for shared drawing and conversation with eye contact*. In: *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 1992, S. 525–532. – URL <http://doi.acm.org/10.1145/142750.142977>. – Zugriffsdatum: 17.02.2007. – ISBN 0-89791-513-5
- [Johanson 2002] JOHANSON, Bradley E.: *Application Coordination Infrastructure for Ubiquitous Computing Rooms*, Department Of Electrical Engineering Of Stanford University, Dissertation, 2002. – URL <http://graphics.stanford.edu/~bjohanso/dissertation/>. – Zugriffsdatum: 16.02.2007
- [Kahlbrandt 2001] KAHLBRANDT, Bernd: *Software-Engineering mit der Unified Modeling Language*. 2. Auflage. Springer, 2001. – ISBN 3-540-41600-5
- [Kruger u.a. 2003] KRUGER, Russell ; CARPENDALE, Sheelagh ; SCOTT, Stacey D. ; GREENBERG, Saul: *How people use orientation on tables: comprehension, coordination and communication*. In: *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*. New York, NY, USA : ACM Press,

- 2003, S. 369–378. – URL <http://doi.acm.org/10.1145/958160.958219>. – Zugriffsdatum: 17.02.2007. – ISBN 1-58113-693-5
- [Neumann 2006] NEUMANN, Carola: *Effizienzsteigerung von Diskussionsprozessen in einem neu gestalteten Konferenzraum*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/neumann.pdf>. – Zugriffsdatum: 22.02.2007
- [O'Reilly 2005] O'REILLY, Tim: *What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software*. 30. September 2005. – URL <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. – Zugriffsdatum: 24.02.2007
- [Raggett u. a. 1999] RAGGETT, Dave ; HORS, Arnaud L. ; JACOBS, Ian: *HTML 4.01 Specification*. 24. Dezember 1999. – URL <http://www.w3.org/TR/html401/>. – Zugriffsdatum: 27.02.2007
- [Roßberger 2007] ROSSBERGER, Philipp: *Physikbasierte Interaktion im Collaborative Workspace*, Hochschule für Angewandte Wissenschaften Hamburg, Seminararbeit, 2007. – URL <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07/rossberger/report.pdf>. – Zugriffsdatum: 27.02.2007
- [Russell u. a. 2002] RUSSELL, Daniel ; TRIMBLE, Jay ; WALES, Roxana: *Two pathes from the same place: Task driven and human-centered evolution of a group information surface*. 2002. – URL <http://ic-www.arc.nasa.gov/publications/pdf/0372.pdf>. – Zugriffsdatum: 21.02.2007
- [Russell u. a. 2005] RUSSELL, Daniel M. ; STREITZ, Norbert A. ; WINOGRAD, Terry: Building disappearing computers. In: *Commun. ACM* 48 (2005), Nr. 3, S. 42–48. – URL <http://doi.acm.org/10.1145/1047671.1047702>. – Zugriffsdatum: 26.02.2007. – ISSN 0001-0782
- [Schrage 1990] SCHRAGE, Michael: *Shared minds: the new technologies of collaboration*. New York, NY, USA : Random House Inc., 1990. – ISBN 0-394-56587-8
- [Scott u. a. 2003] SCOTT, Stacey D. ; GRANT, Karen D. ; MANDRYK, Regan L.: *System Guidelines for Co-located Collaborative Work on a Tabletop Display*. 2003. – URL <http://citeseer.comp.nus.edu.sg/scott03system.html>. – Zugriffsdatum: 25.02.2007
- [Streitz u. a. 1994] STREITZ, Norbert A. ; GEISLER, Jorg ; HAAKE, Jorg M. ; HOL, Jeroen: DOLPHIN: Integrated Meeting Support Across Local and Remote Desktop Environments and LiveBoards. In: *Computer Supported Cooperative Work*, URL <http://citeseer.ist.psu.edu/streitz94dolphin.html>. – Zugriffsdatum: 16.02.2007, 1994, S. 345–358

- [Streitz u. a. 1999] STREITZ, Norbert A. ; GEISLER, Jorg ; HOLMER, Torsten ; KONOMI, Shin'ichi ; MÜLLER-TOMFELDE, Christian ; REISCHL, Wolfgang ; REXROTH, Petra ; SEITZ, Peter ; STEINMETZ, Ralf: i-LAND: An Interactive Landscape for Creativity and Innovation. In: *CHI*, URL <http://citeseer.ist.psu.edu/streitz99iland.html>, 1999, S. 120–127
- [Tandler 2004] TANDLER, Peter: *Synchronous Collaboration in Ubiquitous Computing Environments*, Fachbereich Informatik der Technischen Universität Darmstadt, Dissertation, 2004. – URL <http://elib.tu-darmstadt.de/diss/000506/>. – Zugriffsdatum: 16.02.2007
- [Tanenbaum und van Steen 2002] TANENBAUM, A. S. ; STEEN, M. van: *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2002. – ISBN 0-13-088893-1
- [Weiser 1991] WEISER, Mark: The computer for the twenty-first century. In: *Scientific American* September (1991), S. 94–100
- [Winograd und Flores 1986] WINOGRAD, Terry ; FLORES, Fernando: *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Publishing Corp., 1986. – ISBN 0-89391-050-3

A Anwendungsfalltabellen

Anwendungsfall	Abbruch einer Transaktion	
Zielbeschreibung	Die vom Benutzer getätigten Änderungen im gemeinsamen Bereich sind auf den Stand bei der Übernahme der Transaktion zurückgesetzt.	
Vorbedingung	Der Benutzer hat die Initiative und hat Änderungen an den Inhalten im gemeinsamen Bereich vorgenommen	
Erfolgsbedingung	Der gemeinsame Bereich ist wieder auf dem Ausgangszustand zum Zeitpunkt der Übernahme der Transaktion	
Benutzerrolle	ein Benutzer (hat Initiative)	
Auslöser	Der Benutzer beschließt, dass seine Änderungen abgebrochen und zurückgesetzt werden sollen	
Beschreibung	Schritt	Tätigkeit
	1	Der Benutzer bricht die Transaktion ab
	2	Es wird der Ausgangszustand wiederhergestellt
	3	Der Benutzer kann die Bearbeitung vom Ausgangspunkt der Transaktion aus neu beginnen

Tabelle A.1: Anwendungsfall: Abbruch einer Transaktion

Anwendungsfall	Einfügen von Elementen in den gemeinsamen Bereich	
Zielbeschreibung	Es wurde ein Element aus dem privaten Bereich in den gemeinsamen Bereich kopiert.	
Vorbedingung	Es stehen Elemente im privaten zur Auswahl bereit	
Erfolgsbedingung	Das gewählte Element befindet sich an der gewünschten Stelle im gemeinsamen Bereich.	
Benutzerrolle	mindestens ein Benutzer	
Auslöser	Der Benutzer möchte ein Element in den gemeinsamen Bereich einfügen.	
Beschreibung	Schritt	Tätigkeit
	1	Der Benutzer wählt ein Element aus dem Vorrat in seinem privaten Bereich aus.
	2	Er <i>zieht</i> das Element an die gewünschte Stelle im gemeinsamen Bereich
	3	Das Element wird an der Stelle im gemeinsamen Bereich angezeigt.

Tabelle A.2: Anwendungsfall: Einfügen von Elementen in den gemeinsamen Bereich

Anwendungsfall	Verschieben eines Elements im gemeinsamen Bereich	
Zielbeschreibung	Das gewünschte Element wurde im gemeinsamen Bereich von seinem Ausgangspunkt an die gewünschte Zielposition verschoben.	
Vorbedingung	Es befindet sich mindestens ein Element im gemeinsamen Bereich	
Erfolgsbedingung	Das gewählte Element wurde von seiner Ausgangsposition an die gewünschte Zielposition verschoben.	
Abbruchbedingung	Das Element konnte nicht verschoben werden, da das Ziel keine gültige Zielposition innerhalb des gemeinsamen Bereichs ist.	
Benutzerrolle	mindestens ein Benutzer	
Auslöser	Der Benutzer möchte die Position eines Elements im gemeinsamen Bereich verändern	
Beschreibung	Schritt 1	Tätigkeit Der Benutzer wählt das zu verschiebende Element
	2	Er zieht es an seine neue Position innerhalb des gemeinsamen Bereichs
	3	Das Element befindet sich an seinem Ziel
Alternativen	Schritt 2a	Tätigkeit Das Element wird an eine ungültige Position verschoben
	3a	Das Element befindet sich wieder an seinem Ausgangspunkt

Tabelle A.3: Anwendungsfall: Verschieben eines Elements im gemeinsamen Bereich

Anwendungsfall	Entfernen von Elementen aus dem gemeinsamen Bereich	
Zielbeschreibung	Ein Element wurde aus dem gemeinsamen Bereich entfernt	
Vorbedingung	Es befindet sich mindestens ein Element im gemeinsamen Bereich	
Erfolgsbedingung	Das Element ist aus dem gemeinsamen Bereich entfernt	
Abbruchbedingung	Das Element befindet sich an seiner Ausgangsposition, bzw. an der Position an die es verschoben wurde.	
Benutzerrolle	mindestens ein Benutzer	
Auslöser	Der Benutzer möchte ein Element aus dem gemeinsamen Arbeitsbereich entfernen.	
Beschreibung	Schritt 1	Tätigkeit Der Benutzer wählt ein Element im gemeinsamen Arbeitsbereich aus
	2	Er verschiebt es auf das <i>Löschen-Symbol</i> außerhalb des gemeinsamen Bereichs
	3	Das Element ist aus dem gemeinsamen Bereich entfernt.
Alternativen	Schritt 2a	Tätigkeit Das Element wird an eine ungültige Position verschoben
	3a	Das Element befindet sich wieder an seinem Ausgangspunkt
	2b	Das Element wurde an eine Position innerhalb des gemeinsamen Bereichs verschoben
	3b	Das Element befindet sich an der neuen Position

Tabelle A.4: Anwendungsfall: Entfernen von Elementen aus dem gemeinsamen Bereich

Anwendungsfall	Bereitstellen von Daten für den privaten Bereich	
Zielbeschreibung	Es stehen Daten im privaten Bereich eines Benutzers zur Auswahl bereit	
Vorbedingung	Der Benutzer hat Entsprechende Daten auf einem im Netzwerk erreichbaren Laufwerk hinterlegt	
Erfolgsbedingung	Die Daten wurden erfolgreich bereitgestellt	
Benutzerrolle	mindestens ein Benutzer	
Auslöser	Der Benutzer hat Daten, die er bereitstellen möchte.	
Beschreibung	Schritt	Tätigkeit
	1	Der Benutzer wählt über die private Client-Anwendung die Dateien aus, die Bereitgestellt werden sollen.
	2	Es werden die Daten in das System geladen.
	3	Die Daten stehen dem Benutzer über die Tischanwendung zur Verfügung

Tabelle A.5: Anwendungsfall: Bereitstellen von Daten für den privaten Bereich

Anwendungsfall	Exportieren des erstellten Layouts	
Zielbeschreibung	Die Zeitschrift wird in ein gewähltes Format gespeichert und steht somit für spätere Bearbeitung zur Verfügung.	
Vorbedingung	Es wurde ein Zeitschriftenlayout erstellt, welches abgespeichert werden soll.	
Erfolgsbedingung	Das Layout steht in einer gespeicherten Version bereit.	
Abbruchbedingung	Das Speichern wurde abgebrochen.	
Benutzerrolle	mindestens ein Benutzer	
Auslöser	Der Benutzer möchte das aktuelle Arbeitsergebnis abspeichern.	
Beschreibung	Schritt	Tätigkeit
	1	Auswahl des Speichern-Symbols
	2	Auswahl eines Speicherorts und Dateinamens
	3	Bestätigen des Speichervorgangs
Alternativen	Schritt	Tätigkeit
	2a	Der Benutzer bricht den Speichervorgang ab.
	3a	Der Benutzer bricht den Speichervorgang ab.

Tabelle A.6: Anwendungsfall: Exportieren des erstellten Layouts

Anwendungsfall	Importieren von Layoutbeschreibungen	
Zielbeschreibung	Es soll eine vorhandene Layoutbeschreibung importiert werden.	
Vorbedingung	Es wurde zuvor ein Export des gewünschten Layouts durchgeführt, bzw. es ist eine layoutbeschreibende Datei vorhanden.	
Erfolgsbedingung	Das importierte Layout ist im gemeinsamen Bereich sichtbar.	
Abbruchbedingung	Die zu importierende Layoutbeschreibung konnte nicht erkannt werden.	
Benutzerrolle	mindestens ein Benutzer	
Auslöser	Der Benutzer möchte eine Layoutbeschreibung importieren.	
Beschreibung	Schritt	Tätigkeit
	1	Der Benutzer wählt eine Layoutbeschreibung aus.
	2	Er bestätigt den Import der Datei
	3	Das gewählte Layout wird importiert
	4	Es wird Meldung über erfolgreichen Import angezeigt.
5	Das importierte Layout steht zur Bearbeitung bereit	
Alternativen	Schritt	Tätigkeit
	3a	Der Importvorgang schlägt fehl.
	4a	Es wird eine Meldung über den Fehlschlag angezeigt.
5a	Der gemeinsame Bereich befindet sich im Ausgangszustand.	

Tabelle A.7: Anwendungsfall: Importieren von Layoutbeschreibungen

Anwendungsfall	Austausch von Elementen	
Zielbeschreibung	Ein Benutzer stellt einem anderen Benutzer Elemente aus seinem privaten Bereich zur Verfügung	
Vorbedingung	Mindestens zwei Benutzer am Tisch; der bereitstellende Benutzer hat mindestens ein Element in seinem privaten Bereich	
Erfolgsbedingung	Das übergebene Element steht beiden Benutzern zur Verfügung	
Benutzerrolle	mindestens zwei Benutzer	
Auslöser	Ein Benutzer möchte ein Element einem anderen Benutzer zur Verfügung stellen	
Beschreibung	Schritt	Tätigkeit
	1	Der erste Benutzer zieht das Element in den gemeinsamen Bereich
	2	Der zweite Benutzer zieht das Element aus dem gemeinsamen Bereich in seinen privaten Bereich
	3	Das Element liegt in Kopie bei den Daten des zweiten Benutzers

Tabelle A.8: Anwendungsfall: Austausch von Elementen

Anwendungsfall	Anlegen einer neuen Zeitschriftenseite	
Zielbeschreibung	Es wird eine neue leere Zeitschriftenseite angelegt. Diese befindet sich hinter der angezeigten Seite. Die Anzeige des gemeinsamen Bereichs wechselt auf die neue Seite.	
Vorbedingung	mindestens ein Benutzer	
Erfolgsbedingung	Es ist eine weitere Zeitschriftenseite vorhanden und diese wird auch angezeigt.	
Abbruchbedingung	Der Benutzer bricht das Anlegen der neuen Seite ab.	
Benutzerrolle	ein Benutzer	
Auslöser	Der Benutzer möchte dem Layout eine weitere Seite hinzufügen.	
Beschreibung	Schritt	Tätigkeit
	1	Der Benutzer wählt die Aktion <i>Neue Seite anlegen</i>
	2	Er wird in einem Dialog aufgefordert, die Aktion zu bestätigen.
	3	Er bestätigt die Aktion
	4	Es wird eine neue Seite hinter der aktuellen Seite angezeigt.
5	Es wird die neue Seite im gemeinsamen Bereich angezeigt.	
Alternativen	Schritt	Tätigkeit
	3a	Die Aktion wird über den Dialog abgebrochen.
	4a	Es wird eine Meldung über den Abbruch angezeigt.
5a	Der gemeinsame Bereich befindet sich im Ausgangszustand.	

Tabelle A.9: Anwendungsfall: Anlegen einer neuen Zeitschriftenseite

Anwendungsfall	Entfernen einer Zeitschriftenseite	
Zielbeschreibung	Es wird eine Zeitschriftenseite aus dem Layout gelöscht. Die Anzeige des gemeinsamen Bereichs wechselt auf die folgende Seite.	
Vorbedingung	mindestens ein Benutzer; mindestens eine Layoutseite vorhanden.	
Erfolgsbedingung	Die ausgewählte Seite ist aus dem Layout gelöscht.	
Abbruchbedingung	Der Benutzer bricht das Löschen der Seite ab.	
Benutzerrolle	ein Benutzer	
Auslöser	Der Benutzer möchte eine Seite aus dem Layout entfernen.	
Beschreibung	Schritt	Tätigkeit
	1	Der Benutzer wählt die Aktion <i>Seite löschen</i>
	2	Er wird in einem Dialog aufgefordert, die Aktion zu bestätigen.
	3	Er bestätigt die Aktion
	4	Die Seite wird aus dem Layout entfernt.
5	Es wird die nächste Seite im Layout innerhalb des gemeinsamen Bereichs angezeigt.	
Alternativen	Schritt	Tätigkeit
	3a	Die Aktion wird über den Dialog abgebrochen.
	4a	Es wird eine Meldung über den Abbruch angezeigt.
	5a	Der gemeinsame Bereich befindet sich im Ausgangszustand.
	5b	Es ist keine Seite nach der aktuellen Seite vorhanden.
	6b	Es wird die Seite vor der gelöschten Seite angezeigt.
5c	Es ist nach dem Löschen keine Seite mehr vorhanden.	
6c	Es wird eine neue leere Seite angezeigt.	

Tabelle A.10: Anwendungsfall: Entfernen einer Zeitschriftenseite

Anwendungsfall	Verschieben einer Zeitschriftenseite	
Zielbeschreibung	Es wird eine Zeitschriftenseite an eine neue Position verschoben. Die Anzeige des gemeinsamen Bereichs bleibt auf die verschobene Seite.	
Vorbedingung	mindestens ein Benutzer; mindestens zwei Layoutseiten vorhanden.	
Erfolgsbedingung	Die ausgewählte Seite ist im Layout an eine neue Position verschoben. Die Seite, die vorher an dieser Position war und alle nachfolgenden Seiten werden um eine Position nach hinten verschoben. Alle Seiten, die auf die ursprüngliche Position der verschobenen Seite folgen werden um eine Seite nach vorne verschoben.	
Benutzerrolle	ein Benutzer	
Auslöser	Der Benutzer möchte eine Seite an eine neue Position verschieben.	
Beschreibung	Schritt	Tätigkeit
	1	Es wird die zu verschiebende Seite angezeigt.
	2	Der Benutzer wählt die Aktion <i>Seite verschieben</i>
	3	Er wird in einem Dialog aufgefordert, die neue Position der Seite anzugeben.
	4	Die Seite wird im Layout an die gewünschte Position verschoben.
5	Es wird die verschobene Seite an ihrer neuen Position angezeigt.	
Alternativen	Schritt	Tätigkeit
	3a	Die Aktion wird über den Dialog abgebrochen.
	4a	Es wird eine Meldung über den Abbruch angezeigt.
	5a	Der gemeinsame Bereich befindet sich im Ausgangszustand.

Tabelle A.11: Anwendungsfall: Verschieben einer Zeitschriftenseite

Glossar

AJAX (Asynchronous Javascript and XML) Eine Kombination aus verschiedenen Technologien, die es erlaubt, Webanwendungen zu entwickeln, die sich ähnlich wie Desktopanwendungen verhalten.

Blueboard Ein von IBM entwickeltes Bildschirmsystem, das es erlaubt schnell und unkompliziert auf persönliche Daten zuzugreifen.

BumpTop Der Prototyp eines Computerdesktops, der mehr an der natürlichen Schreibtischmetapher orientiert.

Clearboard Eine in [Ishii und Kobayashi \(1992\)](#) vorgestellte frühe Technologie, die das entfernte kooperierende Arbeiten erleichtern soll.

Collaborative Workspace *hier:* Ein technologiegestützter Konferenzraum, der die Effizienz des gemeinsamen Arbeitens erhöhen soll.

CommChair Eine vom Fraunhofer IPSI entwickelte Kombination Stuhl, Bildschirm und PC für den Einsatz im Collaborative Workspace.

Connctable Eine vom Fraunhofer IPSI entwickelte Kombination aus Bildschirm und PC für den Einsatz im Collaborative Workspace. Die Besonderheit des Connctable liegt darin, dass sich die Bildschirme zweier Geräte dieser Art verbinden können und ihren Benutzern einen größeren gemeinsamen Bildschirmausschnitt bieten.

CSCW (Computer Supported Collaborative Work) Mit CSCW werden allgemein die grundlegenden Technologien und Methodologien bezeichnet, die die computergestützte Zusammenarbeit fördern.

Digitales Whiteboard Eine digitale Wandtafel. Häufig repräsentiert durch einen berührungsempfindlichen Bildschirm.

Gecko-Rendering-Engine Gecko ist ein Programmmodul zur Darstellung des HTML-Codes von Webseiten. Es unterstützt eine Reihe offener Standards, wie z. B. HTML, XHTML, SVG, CSS.

Group Space Mit *Group Space* wird der gemeinsame Arbeitsbereich einer Groupware-Anwendung bezeichnet. In diesem Bereich werden die Ergebnisse der erstellt.

HCI (Human-Computer-Interaction) Ein Teilgebiet der Informatik, das sich mit der benutzergerechten Gestaltung von Informationssystemen Beschäftigt. (deutsch: *Mensch-Computer-Interaktion*)

Initiative *Im Rahmen dieser Arbeit* bezeichnet der Begriff *Initiative* den Zustand eines Benutzers, der momentan hauptsächlich an einer gemeinsamen Aufgabe arbeitet. Auf Tabletop-Arbeitsflächen ist dieses zusätzlich derjenige, dem der gemeinsame Arbeitsbereich zugewendet ist.

Initiativewechsel *hier:* Die Weitergabe der *Initiative* an einen anderen Benutzer

IPSI Institut für Integrierte Publikations- und Informationssysteme der Fraunhofer-Gesellschaft

ISBN (International Standard Book Number) Ein international eindeutiger Bezeichner für Buchveröffentlichungen

Personal Space Mit *Personal Space* wird der persönliche Arbeitsbereich einer Groupware-Anwendung bezeichnet. In diesem Bereich werden die jeweiligen Vorarbeiten der Gruppenteilnehmer erstellt.

physikbasierte Interaktion *hier:* Interaktion auf einem Bildschirm, bei der den dargestellten Elementen physikalische Eigenschaften, wie z. B. Gewicht und Reibung gegeben werden.

REST (Representational State Transfer) ein zustandsloses Client-/Server-Protokoll, welches eine Reihe wohldefinierter Operationen erlaubt.

RFID (Radio Frequency Identification) RFID ermöglicht eine berührungslose Identifikation von Objekten.

Rich-Web-Client Ein besonderer Web-Client, der gegenüber normalen Web-Clients eine Reihe von Funktionalitäten übernimmt, die sonst auf dem Server ausgeführt würden.

SVG (Scalable Vector Graphics) ein Standard zur Beschreibung von 2D-Vektorgrafiken.

Tabletop Die spezielle Ausrichtung einer Arbeitsfläche, ähnlich der eines Tisches

Ubiquitous Computing Die Vision des allgegenwärtigen Computers. Teil dieser Vision ist es, dass die Computer zwar allgegenwärtig sind. Sie treten allerdings auf Grund ihrer Form und Funktion so in den hintergrund, dass sie nicht mehr als solche wahrgenommen werden. ([Weiser, 1991](#))

WEB 2.0 Eine Reihe von Thesen, die die Vision des zukünftigen World Wide Web beschreiben. Formuliert wurden diese Thesen in [O'Reilly \(2005\)](#).

Whiteboard Eine Wandtafel, mit einer glatten weißen Oberfläche. Sie ist beschreibbar mit speziellen Filzmarkern. Ein Whiteboard wird häufig zur Ideenfindung und zur Verdeutlichung von Strukturen benutzt. Die beschriebene Fläche lässt sich leicht wieder reinigen.

Index

AJAX, [13](#), [14](#), [35–38](#), [41](#), [47](#), [54](#), [58](#), [61](#)

ajax

-Presents, [39](#)

-Sketch, [39](#)

-Write, [39](#)

-XLS, [39](#)

Ajax13, [11](#), [39](#)

Architektur, [33](#)

Augenkontakt, [45](#)

Backend, [51](#)

Benutzeroberfläche, [39](#)

Benutzerschnittstelle, [33](#)

BitTorrent, [35](#)

BlueBoard, [25](#), [46](#)

BumpTop, [31](#)

ClearBoard, [24](#)

Collaborative Workspace, [15](#)

CommChair, [20](#)

ConnecTable, [18](#)

ConnectTable, [20](#)

Crosscompiler, [61](#)

Desktopanwendung, [33](#), [37](#), [38](#), [40](#)

digitales Whiteboard, [23](#)

Disappearing Computer, [13](#)

Drag-and-Drop, [44](#)

DynaWall, [20](#), [23](#), [24](#)

Einzelplatzanwendung, [40](#)

Element, [58](#)

ewiges Beta, [36](#)

Flickr, [59](#)

FTP-Client, [59](#)

Gecko, [54](#), [55](#)

gemeinsamer Bereich, [53](#)

geräteunabhängig, [36](#)

Geräteunabhängigkeit, [40](#)

Geschäftlogik, [55](#)

Gesichtsmimik, [12](#)

Gesten, [12](#)

Google, [35](#)

Calendar, [11](#)

Docs, [39](#), [40](#)

Earth, [38](#)

Mail, [11](#)

Maps, [38](#)

Webtoolkit, [61](#)

Group Space, [29](#), [30](#), [44](#)

Groupware, [13](#), [51](#)

HTML, [55](#)

HTTP-Protokoll, [55](#)

Infrastruktur, [33](#)

Initiative, [29](#), [30](#)

Inititive

-wechsel, [29](#)

InteracTable, [28](#), [54](#)

Internet als Plattform, [35](#)

iRoom, [20](#), [21](#), [23](#)

iTable, [28](#)

Java, [59](#), [61](#)

JavaScript, [61](#)

- Javascript, [54](#)
- Körperhaltung, [12](#)
- Kommunikation, [40](#)
 - entfernt, [12](#)
- Konferenzraum, [54](#)
- kooperierendes Arbeiten, [22](#)
- Lightweight Programming Models, [36](#)
- Meebo, [11](#), [40](#)
- Meeting, [12](#)
- Microsoft Office, [39](#)
- MindMapping, [38](#), [40](#)
- Nutzung kollektiver Intelligenz, [35](#)
- OpenOffice, [39](#)
- Orientation
 - Environment-based automatic, [30](#)
 - Fixed, [30](#)
 - Manual, [30](#)
 - Person-based automatic, [30](#)
- Pairware, [13](#), [51](#)
- Personal Space, [29](#), [44](#)
- Puzzle, [30](#)
- REST, [36](#)
- RFID, [25](#)
- Rich User Experiences, [36](#)
- Rich-Web-Client, [58](#)
- Rich-Web-Clients, [61](#)
- Roomware, [15](#), [18](#)
- SOAP, [36](#)
- Software
 - Lebenszyklus, [36](#)
 - als Dienst, [36](#)
- SVG, [55](#), [60](#)
- Tabletop, [28](#), [33](#), [47](#)
- Tabletop-Anwendung, [45](#)
- Thin-Client, [25](#)
- Tim O'Reilly, [35](#)
- Touchscreen, [54](#)
- Transaktionsverwaltung, [45](#)
- Ubiquitous Computing, [22](#)
- User Centered Design, [13](#), [46](#)
- virtuelle Maschine, [11](#)
- Vorhalten spezialisierter Daten, [35](#)
- WEB 2.0, [35–37](#), [40](#), [55](#), [62](#)
- Web-Anwendung, [37](#)
- Webanwendung, [12](#), [37](#), [40](#)
- Webbrowser, [37](#)
- Webserver, [11](#)
- Whiteboard, [12](#)
- Zeitschrift, [58](#)

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 27. Februar 2007

Ort, Datum

Unterschrift