



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Diplomarbeit

Michael Gottwald

Untersuchung von Interaktionsmöglichkeiten bei  
synchroner computergestützter Gruppenarbeit in  
heterogener Systemumgebung

Michael Gottwald

Untersuchung von Interaktionsmöglichkeiten bei  
synchroner computergestützter Gruppenarbeit in  
heterogener Systemumgebung

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

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

Abgegeben am 28. November 2007

## **Michael Gottwald**

### **Thema der Diplomarbeit**

Untersuchung von Interaktionsmöglichkeiten bei synchroner computergestützter Gruppenarbeit in heterogener Systemumgebung

### **Stichworte**

Software-Ergonomie, Ubiquitous Computing, CSCW, Groupware, Mensch-Computer-Interaktion

### **Kurzzusammenfassung**

Diese Arbeit beschäftigt sich mit neuen Interaktionsformen in einem computergestützten Konferenzraum. Zukünftige Konferenzräume bestehen nicht nur aus einzelnen Computergeräten, die jedes für sich, beschränkt auf dessen Aufgabenbereich, den Menschen unterstützen. Sie enthalten vielmehr zusammenarbeitende vernetzte Komponenten, die den Menschen bei seiner kooperativen Bearbeitung von Aufgaben helfen sollen. Die hier untersuchte Interaktion bezieht sich auf die entfernte Steuerung und Integration von größeren Ein- und Ausgabegeräten (z.B. Powerwalls) zur Verfügung gestellten virtuellen Arbeitsflächen (Desktops). Das für diese Untersuchung eingesetzte Verfahren ist das „Prototyping“: Mit Hilfe eines Prototyps sollen die analysierten und neu konzipierten Interaktionen überprüft werden.

## **Michael Gottwald**

### **Title of the paper**

Study of interaction possibilities for synchronous computer-supported groupwork in heterogeneous system environment

### **Keywords**

Software-Ergonomics, Ubiquitous Computing, CSCW, Groupware, Human-Computer-Interaction

### **Abstract**

This work deals with new forms of interaction in a computer supported conference room. Future conference rooms do not consist only of separate computers individually supporting people in their respective areas of responsibility. Rather they consist of components linked up in a co-operating network that helps people working cooperatively with their tasks. The interaction to be examined in this work is remote control and integration of desktops being put at disposal by input/output devices, e.g. powerwalls. The method used here is prototyping: With the aid of a prototype the analysed and newly designed interaction is to be checked.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>8</b>
<b>Abbildungsverzeichnis</b>	<b>9</b>
<b>1. Einleitung</b>	<b>13</b>
1.1. Zielsetzung, Methodik und Abgrenzung . . . . .	15
1.2. Inhalt und Gliederung der Arbeit . . . . .	16
<b>2. Theoretische Grundlagen</b>	<b>18</b>
2.1. Computer Supported Cooperative Work . . . . .	18
2.1.1. Begriffe und Definitionen . . . . .	18
2.1.2. Motivation für den Einsatz von CSCW . . . . .	19
2.1.3. CS(CW) - Cooperative Work . . . . .	20
2.1.4. CSCW-Systeme . . . . .	22
2.2. HCI . . . . .	24
2.2.1. Motivation und Zukunft . . . . .	25
2.2.2. HCI und CSCW-Systeme . . . . .	26
2.2.3. Abgrenzung - CHI, MMI . . . . .	26
2.3. Ubiquitous Computing . . . . .	26
2.3.1. UbiComp versus Virtual Reality . . . . .	27
2.3.2. Themenspektrum von UbiComp . . . . .	27
2.3.2.1. Entwurf neuer Interaktionen . . . . .	27
2.3.2.2. Privatsphäre, Vertrauen und Sicherheit . . . . .	28
2.3.2.3. Infrastruktur . . . . .	28
2.3.2.4. Kontext . . . . .	28
2.3.2.5. Informationsfindung . . . . .	29
2.3.3. UbiComp-Software-Infrastruktur und CSCW . . . . .	29
2.4. Software-Ergonomie . . . . .	31
2.4.1. Ergonomie allgemein . . . . .	31
2.4.2. Software-Ergonomie . . . . .	32
2.4.3. CSCW-Systeme und Software-Ergonomie . . . . .	32
2.4.3.1. Kontext . . . . .	35
2.4.3.2. Kontrolle . . . . .	36

---

2.4.3.3. Dialoggestaltung . . . . .	37
2.4.4. Usability . . . . .	41
2.5. CSCW-System - Collaborative Workspace - Ein Beispiel . . . . .	41
2.5.1. Roomware - Ein Überblick . . . . .	42
2.5.2. Roomware - UbiComp . . . . .	42
2.5.3. Roomware - HCI . . . . .	43
<b>3. Analyse</b>	<b>53</b>
3.1. Rahmenbedingungen und Abgrenzung . . . . .	54
3.1.1. Umgebung . . . . .	54
3.1.2. Mittel . . . . .	56
3.1.3. Grenzen der Bewertung . . . . .	56
3.2. Szenarien . . . . .	56
3.2.1. Szenarien eines Software-Entwicklungsprozesses . . . . .	57
3.2.2. Test-Szenario . . . . .	60
3.2.2.1. Szenariobeschreibung . . . . .	61
3.2.3. Anwendungsfälle . . . . .	61
3.3. Die Idee . . . . .	62
3.3.1. Kontrolle und Steuerung entfernter Geräte . . . . .	63
3.3.1.1. Virtual Network Computing: . . . . .	64
3.3.1.2. Gemeinsamer Arbeitsplatz/Desktop: . . . . .	65
3.3.2. Multi-Desktop-Steuerung . . . . .	68
3.3.2.1. Multi-Desktop-Steuerung: Gesamtbild . . . . .	68
3.3.2.2. MDS - Dokumente verschieben . . . . .	69
3.3.2.3. MDS - Verschieben geöffneter Anwendungen . . . . .	70
3.3.2.4. MDS - Dokumente verschieben und öffnen . . . . .	71
3.3.2.5. MDS - Anmeldung . . . . .	72
3.3.2.6. MDS - Verteilung von Dokumenten . . . . .	72
3.3.3. MDS - Grenzen der Realisierung . . . . .	73
<b>4. Prototyp - „ProjectRoom“</b>	<b>76</b>
4.1. Entwurf und Realisierung . . . . .	77
4.1.1. Architektur des Systems . . . . .	77
4.1.2. Room-Client und Room-Server . . . . .	79
4.1.3. Room-Services . . . . .	80
4.1.4. Kommunikation . . . . .	81
4.1.5. Service-Verwaltung . . . . .	81
4.1.6. Entwurfsentscheidungen . . . . .	81
4.1.6.1. Kommunikation . . . . .	81
4.1.6.2. Programmiersprache und Plattform . . . . .	82
4.1.6.3. Hardware und Einsatz . . . . .	82

---

4.1.7. Sequenzdiagramme . . . . .	83
4.1.7.1. Systemanmeldung und Service-Bereitstellung . . . . .	83
4.1.7.2. Kopieren und Bearbeiten eines Dokumentes . . . . .	84
4.2. Implementierung . . . . .	85
4.2.1. ProjectRoomClientApp . . . . .	86
4.2.1.1. Initialisierte Anwendung . . . . .	86
4.2.1.2. Anwendung in angemeldetem Zustand . . . . .	87
4.2.1.3. Toolbox . . . . .	88
4.2.1.4. Verteilte Toolboxes . . . . .	89
4.2.2. ProjectRoomServer . . . . .	89
4.2.2.1. Initialisierte Anwendung . . . . .	89
4.2.2.2. Server-Anwendung - Service gestartet . . . . .	91
4.2.2.3. Verteilte Toolboxes beim Server . . . . .	92
4.2.3. Interaktionen . . . . .	93
4.2.3.1. Benutzer-Anmeldung . . . . .	93
4.2.3.2. Dokument transferieren und Initiierung eines Bearbeitungs- prozesses . . . . .	93
4.2.3.3. Verteilen von Dokumenten . . . . .	94
4.2.4. Konfiguration/Administration der Anwendungen . . . . .	95
4.2.4.1. Konfiguration von ProjectRoomClientApp . . . . .	95
4.2.4.2. Administration von ProjectRoomServer . . . . .	96
4.3. Software-Technisches Fazit . . . . .	97
<b>5. Fazit</b>	<b>102</b>
5.1. Beurteilung des Prototyps . . . . .	102
5.2. Ausblick . . . . .	104
<b>Literaturverzeichnis</b>	<b>105</b>
<b>A. Tabellen der Anwendungsfälle</b>	<b>107</b>
<b>B. Prototyp-Abbildungen</b>	<b>112</b>
B.1. Abbildungen: ProjectRoomClientApp . . . . .	112
B.1.1. Abbildungen: Toolbox . . . . .	114
B.2. Abbildungen: ProjectRoomServer . . . . .	116
B.3. Abbildungen: Interaktionen . . . . .	119
B.3.1. Benutzer-Anmeldung . . . . .	119
B.3.2. Dokument transferieren und Initiierung eines Bearbeitungsprozesses .	121
B.3.3. Verteilen von Dokumenten . . . . .	125
B.3.4. Zusatz - Konfiguration/Administration der Anwendungen . . . . .	127
B.3.4.1. Konfiguration von ProjectRoomClientApp . . . . .	127

---

B.3.4.2. Administration von ProjectRoomServer . . . . . 129

# Tabellenverzeichnis

2.1. Anforderung an einer Software-Infrastruktur in ubiquitären Umgebungen (aus Tandler (2004)) . . . . .	30
2.2. Grundsätze der Dialoggestaltung nach DIN 9241-110 . . . . .	33
A.1. Anwendungsfall 1: Systemanmeldung . . . . .	107
A.2. Anwendungsfall 2: Bereitstellung . . . . .	108
A.3. Anwendungsfall 3: Verteilung . . . . .	109
A.4. Anwendungsfall 4: Initiierung eines Bearbeitungsprozesses . . . . .	110
A.5. Anwendungsfall 5: Administration des Servers . . . . .	111
A.6. Anwendungsfall 6: Konfiguration des Clients . . . . .	111

# Abbildungsverzeichnis

1.1. Drag-And-Throw: (A) Objekt wählen, (B) Bogen aktivieren, (C) Bogen spannen, bis gewünschte Weite erreicht ist, (D) Objekt zur gewünschten Position werfen . . . . .	15
2.1. Einzelne Forschungsbereiche . . . . .	44
2.2. Leavitt-Raute: Wechselwirkungen zwischen den Komponenten: Mensch, Organisation, Arbeit, Technik - CSCW-System beziehen sich vorwiegend auf die Technik-Komponente . . . . .	44
2.3. Kommunikationsgrad innerhalb einer Gruppe . . . . .	45
2.4. Kooperative Arbeit: Fokus und Planen (nach Schlichter) . . . . .	45
2.5. Raum-Zeit-Matrix: Mögliche Anwendungen (Groupware), abhängig von Raum und Zeit . . . . .	46
2.6. Raum-Zeit-Matrix nach (?): Zusätzliche Unterteilung in Vorhersehbarkeit bei getrennt räumliches/zeitliches gemeinsames Arbeiten . . . . .	46
2.7. 3K-Pyramide - Kommunikations-, Koordinations-, Kooperationsunterstützung einzelner Systeme . . . . .	47
2.8. Forschungsthemen und dessen Beziehung zueinander . . . . .	48
2.9. Haupt-Trends in Computing . . . . .	49
2.10. Abgewandelte Leavitt-Raute im CSCW-Kontext . . . . .	49
2.11. Zusammenhang zwischen Arbeitspsychologischen Kontrollkonzept und den Grundsätzen ergonomischer Dialoggestaltung . . . . .	50
2.12. Usability - Design einer Teekanne (aus „The Design of Everyday Things“ von Norman (2002)) . . . . .	50
2.13. Zeigt unter Berücksichtigung der BEACH-Architektur die Roomware-Komponenten	51
2.14. Roomware-Projekt: Überblick . . . . .	52
3.1. CSCW-Raum in der HAW-Hamburg . . . . .	55
3.2. Autor-Kritiker-Zyklus beim WAM-Ansatz, (nach Neuman, 2006, S. 10) . . . . .	57
3.3. Aufbau und Abläufe im Workflow (nach Neuman, 2006, S. 30) . . . . .	59
3.4. Interaktiver Gruppenarbeitsraum - Skizze . . . . .	63
3.5. Virtual Network Computing . . . . .	64

3.6. Gemeinsamer Desktop/Arbeitsplatz - Darstellung und Interaktion: (1)-(3) mögliche Übergänge, (4a) u. (4b) Bewegung eines Objekts v. Desktop 2 zu Desktop 1 . . . . .	67
3.7. Gemeinsamer Desktop/Arbeitsplatz - Navigation: (A)-(D) stellt eine Powerwall dar, (F) ist der Original Desktops des Anwenders, (E) Fensterausschnitt des gemeinsamen Desktop, (1) möglicher Übergang Original-Desktop/Powerwall-Desktop . . . . .	68
3.8. Bumping Displays 1 - (a) Kacheln dynamischer Displays durch das Zusammenstoßen (bumping) zweier Tablets. (b) Die Tablets bilden zeitweilig ein größeres Display und ermöglichen dadurch die Anzeige eines größeren Fotos (Gebirge). Die kleinen roten Pfeile signalisieren dem Benutzer (Feedback) die Verbindung beider Tablets bei diesen Kanten. . . . .	69
3.9. Bumping Displays 2 - (a) Hier sieht man eine Face-To-Face (von Angesicht zu Angesicht) Kollaboration durch das Zusammenführen beider Tablets. Die abgebildete Skizze (Haus) mit Notizen der Benutzer wird korrekt ausgerichtet. (b),(c) Die Benutzer bekommen eine klare Rückmeldung in Form von aussagekräftigen Piktogrammen über die Verbindung oder spätere Unterbrechung der kollaborativen Sitzung. . . . .	70
3.10.MDS - Gesamtbild: (a) Entfernte Desktop, (b) Geöffnete Anwendungen, (c) Dokumente . . . . .	71
3.11.MDS - Verschieben 1: Verschieben eines Dokumentes vom lokalen Desktop (Laptop) zu einem entfernten Desktop (Display X). . . . .	72
3.12.MDS - Verschieben 2: Verschieben einer geöffneten Anwendung . . . . .	73
3.13.MDS - Verschieben 3: Verschieben eines Dokumentes und Öffnen mit einer adäquaten Anwendung bei dem Ziel-Desktop. . . . .	74
3.14.MDS - Anmeldung 1: Zwei Benutzer möchten sich bei einem entfernten Desktop anmelden. . . . .	74
3.15.MDS - Anmeldung 2: Die Benutzer sind angemeldet. . . . .	75
3.16.MDS - Verteilung: Teilnehmer 1 sendet ein Dokument an Teilnehmer 2. . . . .	75
4.1. Entwurf - Client/Server-Architektur mit Service-Bus/Verwaltung . . . . .	78
4.2. Entwurf . . . . .	79
4.3. Model-View-Controller der einzelnen Komponenten . . . . .	80
4.4. Entwurf - Client MVC-Architektur . . . . .	80
4.5. Realisierung - Verteilung der Komponenten, Deployment . . . . .	82
4.6. Sequenzdiagramm 1 - Service-Bereitstellung und Systemanmeldung . . . . .	84
4.7. Sequenzdiagramm 2 - Kopieren eines Dokumentes auf den Server und anschließender Bearbeitung durch eine Anwendung . . . . .	85
4.8. ProjectRoomClientApp - Gestartete Anwendung . . . . .	86
4.9. ProjectRoomClientApp - Der Benutzer hat sich angemeldet . . . . .	88

---

4.10. Toolbox - Freier und eingebetteter Modus . . . . .	89
4.11. ProjectRoomClientApp - Die Toolboxen liegen (frei verschiebbar) auf dem lokalen Desktop . . . . .	90
4.12. ProjectRoomServer - Gestartete Anwendung . . . . .	91
4.13. ProjectRoomServer - Gestartete Anwendung und Service . . . . .	92
4.14. ProjectRoomServer - Verteilte Toolboxen auf dem Server-Desktop . . . . .	92
4.15. Benutzer-Anmeldung - Dialog zur Anmeldung . . . . .	94
4.16. Benutzer-Anmeldung - Inhalt der Benutzer-Toolboxen . . . . .	95
4.17. Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop der Powerwall vor der Ausführung . . . . .	96
4.18. Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers und der Verschiebevorgang . . . . .	97
4.19. Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Anzeige des Bildes mit einer Anwendung (IrfanView), aktualisierte Dokumente-Toolbox . . . . .	98
4.20. Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers nach der Aktion . . . . .	99
4.21. Verteilen von Dokumenten - Die Desktops der beiden Benutzer vor der Verteilung . . . . .	100
4.22. Verteilen von Dokumenten - Das Dokument taucht in der Dokumente-Toolbox von „Buckaroo“ auf. . . . .	100
4.23. Konfiguration von ProjectRoomClientApp - NameSerververbindungs- und Benutzereinstellungen . . . . .	101
4.24. Administration von ProjectRoomServer - Sämtliche Dialog-Seiten . . . . .	101
B.1. ProjectRoomClientApp: Gestartete Anwendung . . . . .	112
B.2. ProjectRoomClientApp - Der Benutzer hat sich angemeldet . . . . .	113
B.3. Toolbox - Freier und eingebetteter Modus . . . . .	114
B.4. ProjectRoomClientApp - Die Toolboxen liegen (frei verschiebbar) auf dem lokalen Desktop . . . . .	115
B.5. ProjectRoomServer - Gestartete Anwendung . . . . .	116
B.6. ProjectRoomServer - Gestartete Anwendung und Service . . . . .	117
B.7. ProjectRoomServer - Verteilte Toolboxen auf dem Server-Desktop . . . . .	118
B.8. Benutzer-Anmeldung - Dialog zur Anmeldung . . . . .	119
B.9. Benutzer-Anmeldung - Inhalt der Benutzer-Toolboxen . . . . .	120
B.10. Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop der Powerwall vor der Ausführung . . . . .	121
B.11. Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers und der Verschiebevorgang . . . . .	122

---

B.12.Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Anzeige des Bildes mit einer Anwendung (IrfanView), aktualisierte Dokumente-Toolbox . . . . .	123
B.13.Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers nach der Aktion . . . . .	124
B.14.Verteilen von Dokumenten - Die Desktops der beiden Benutzer vor der Verteilung . . . . .	125
B.15.Verteilen von Dokumenten - Das Dokument taucht in der Dokumente-Toolbox von „Buckaroo“ auf. . . . .	126
B.16.Konfiguration von ProjectRoomClientApp - NameServer-Verbindungseinstellungen	127
B.17.Konfiguration von ProjectRoomClientApp - Benutzereinstellungen . . . . .	128
B.18.Administration von ProjectRoomServer - NameServer-Verbindungseinstellungen	129
B.19.Administration von ProjectRoomServer - Verwalten der registrierten Anwendungen . . . . .	130
B.20.Administration von ProjectRoomServer - Remote-Desktop-Einstellungen . . .	131
B.21.Administration von ProjectRoomServer - Benutzereinstellungen . . . . .	132
B.22.Administration von ProjectRoomServer - Registrierung neuer Anwendungen .	133

# 1. Einleitung

Man stelle sich Folgendes vor:

Ein Projektleiter eines Software-Unternehmens, hat für den heutigen Tag eine Besprechung des aktuellen Projektes angesetzt. Ziel ist es, den aktuellen Projektfortschritt aufzuzeigen sowie weitere Aufgaben zu besprechen. Das Projekt selbst, ist eine neue E-Learning-Plattform, die speziell für mittelständische Unternehmen, konzipiert wird. Das Projekt besteht aus Teilbereichen entsprechend softwaretechnischer Disziplinen:

- Datenpersistenz
- Benutzerschnittstelle/Ergonomie/Usability
- Sicherheit
- Server-Architektur
- Client-Architektur

Der Projektleiter erscheint als erstes vor dem Projekt-Gruppenraum und registriert sich mittels ID-Card und einer speziellen Geräteeinheit (bestehend aus einem Touch-Screen und Card-Reader). Als authentifizierter Benutzer wählt er das betreffende Projekt in einer Projekt-Auswahlliste, betritt den Raum und sieht die schon initialisierten Anzeigegerät mit dem aktuellen Projekthalt (Projektstruktur o.ä.). Der Raum ist mit diversen Anzeige-Geräten unterschiedlicher Größe (z.B. mehrere Meter umfassenden Wandbildschirm), Touch-Screens<sup>1</sup> (an den Wänden und auch eingelassen in Tische), sowie mobilen Arbeitsplätzen (bestehend aus PC und Anzeigeeinheit) ausgestattet.

Nach und nach treffen die Projektgruppen respektive -teilnehmer ein. Die Anmeldung einzelner mitgeführter Hardware (Notebook, Handheld, Mobile) erfolgt beim Betreten des Raumes automatisch. Natürlich stehen auch die stationären Arbeitsplätze - bereits initialisiert mit Projektdaten - für jeden Projektteilnehmer zur Verfügung. Einige Teilnehmer können bei der Besprechung nicht anwesend sein, und melden sich daher via Internet an den Projektraum

---

<sup>1</sup>Berührungssensitive Bildschirme, welche mittels Finger oder Stifte bedient werden

an. Auf einen der Anzeige-Geräte werden sämtliche angemeldete (externe/interne) Benutzer symbolisch, als Avatar <sup>2</sup>, angezeigt.

Der Projektleiter eröffnet die Besprechung und bittet jeden Gruppenleiter kurz über den Fortschritt des Teilprojektes zu berichten. Der Vortragende kann hierbei sämtliche Anzeigegeräte benutzen - sowohl von seinem Platze als auch von dem Podium aus. Die gezeigten Informationen können von jedem Teilnehmer lokal kommentiert und später zur Diskussion gestellt werden. Es stellt sich heraus, dass der Zeitplan aller Voraussicht nach nicht eingehalten werden kann. Der Projektleiter bittet jede Gruppe, sich Gedanken zu machen, wo in ihren Teilbereichen noch Zeit eingespart werden könnte. Die einzelnen Gruppen diskutieren in unterschiedlichen Bereichen des Raumes, wobei alle zur Verfügung stehenden Geräte (eigen oder „Raumgeräte“) benutzt werden können. Benötigte Daten (Dokumente, Quell-Code, Diagramme, etc.) können synchron angefordert respektive bearbeitet werden. Nach Ablauf der hierfür vorgesehenen Zeit, eröffnet der Projektleiter abermals die Sitzung. Alle Veränderungen, die sich während der Besprechung ergeben, werden automatisch gesichert und protokolliert.

Dieser kurz vorgestellte Ablauf einer Projektbesprechung mag noch eine Vision sein - aber wie lange noch? Im relativ neuen Informatik-Zweig „Computer-Supported Cooperative Work“ (CSCW), werden gerade Techniken, sowohl software-technischer als auch hardware-technischer Art, untersucht und konzipiert, die eine derartige Zusammenarbeit von Menschen, gestützt durch Computer, ermöglichen soll. Dieser Bereich berührt unmittelbar andere Themengebiete auf die ich später noch kurz eingehen werde, wie z. B.:

1. Human Computer Interaction (HCI)
2. Ubiquitous Computing (UbiComp)
3. Collaborative Workspace (CWS)
4. Usability
5. Ergonomie

Die in einem CSCW-Raum zum Teil neu eingesetzten Techniken (wie z. B. große berührungssensitive Displays, TableTop-Displays) bedürfen auch neuer Interaktionstechniken respektive -Schemata. Das Arbeiten mit großen Displays und dementsprechend großer Arbeitsflächen mag hier schon andere Anforderungen an die zugehörige Benutzerschnittstelle stellen, als bei einem Einzelplatzsystem (PC). Als Beispiel sei hier die Drag-And-Throw-Technik von (Collomb und Hascoet (2004)) genannt, die Objekte von einer Position mittels eines „Bogens“ in weiter entfernten Desktop-Regionen „schießt“

---

<sup>2</sup>Ein Avatar ist eine künstliche Person oder ein grafischer Stellvertreter einer echten Person in der virtuellen Welt.



Abbildung 1.1.: Drag-And-Throw: (A) Objekt wählen, (B) Bogen aktivieren, (C) Bogen spannen, bis gewünschte Weite erreicht ist, (D) Objekt zur gewünschten Position werfen

Diese Arbeit bezieht sich u. a. auf die Diplomarbeit von Carola Neumann (Neuman (2006)) „Effizienzsteigerung von Diskussionsprozessen in einem neu gestalteten Konferenzraum“. Eines der Ziele war,

die Anforderungen für ein Konferenzsystem zu analysieren und daraus ein Konzept bzw. einen Prototypen für die Benutzungsschnittstelle des Systems zu erarbeiten.

In Kapitel 3 gehe ich noch detaillierter auf die Arbeit von Neumann ein.

## 1.1. Zielsetzung, Methodik und Abgrenzung

Die Diplomarbeit bewegt sich, wie aus der Einleitung zu ersehen ist, im Bereich synchrones computer-unterstütztes kooperatives Arbeiten. Sie soll u. a. die Frage beantworten, wie Benutzer in einem CWS interagieren können - hierfür werden vier grundlegende Anwendungsfälle herausgegriffen:

1. Anmeldung an das CSCW-System
2. Bereitstellung von Artefakte
3. Verteilung von Artefakte
4. Initiierung eines Bearbeitungsprozess

Diese Arbeit vergleicht und analysiert mögliche Interaktionsschemata in einem CSCW-System. Hierbei wird insbesondere Bezug genommen auf die in Neumanns Diplomarbeit herausgearbeiteten Szenarien eines Software-Entwicklungsprozesses. Wie könnte eine Zusammenarbeit in einen CWS aussehen? Welche Interaktionsmöglichkeiten sind bei einer koordinierten Benutzung mehrerer Ein- und Ausgabe-Geräte vorstellbar? Und, vor allen Dingen, sind diese in bestimmten Szenarien brauchbar?

Diese Fragen sollen anhand eines Prototyps beantwortet werden. Zuvorderst wird das CSCW und dessen angrenzenden respektive überschneidenden Bereiche detaillierter beschrieben. Hauptaugenmerk wird hierbei auf HCI (Human Computer Interaction) gelegt. Nachfolgend werden grundlegende Interaktionsarten in einen CWS gezeigt und analysiert, unterschiedliche Sichtweisen auf ein CSCW-Raum beschrieben sowie auf die Diplomarbeit von (Neuman (2006)) eingegangen - insbesondere auf die dort aufgezeigten Szenarien. Des Weiteren werden die zuvor untersuchten Szenarien und die dafür infrage kommenden Interaktionsschemata analysiert. Natürlich darf eine Beschreibung des Prototyps nicht fehlen (Architektur, Design, Realisierung). Zu guter Letzt erfolgt das obligatorische Fazit, welches den Prototyp konzeptuell, unter Berücksichtigung der software-ergonomischen Prinzipien für CSCW-System, bewertet und die Ziele den erarbeiteten Ergebnissen gegenüberstellt.

Der hier im Rahmen der Untersuchung benutzte Prototyp stellt keine Realisierung eines vollständigen CSCW-Systems dar, sondern beschäftigt sich nur mit einem Teilaspekt in einen CSCW-Raum. Die üblichen Entwicklungsphasen Analyse, Design und Realisierung werden nicht in aller Ausführlichkeit beschrieben. Detailliertere Architekturbetrachtungen sind bitte der Diplomarbeit von Roman Bartnik „Weiterentwicklung einer Technologiebasis für interaktive Gruppenarbeitsräume“ (Bartnik (2006)) zu entnehmen. Sicherheitsrelevante Betrachtungen wie Authentifizierung, Rechteverwaltung, Rechtestrukturen sowie der Entwurf eines adäquaten Modells zur Datenpersistenz in einem CSCW-System sind in der Diplomarbeit „Berechtigungsstrukturen in kollaborativen Umgebungen“ von (Horst Mund (2006)) zu finden.

## 1.2. Inhalt und Gliederung der Arbeit

Nachfolgend sei eine kurze Übersicht der Kapitel gegeben:

### Kapitel 2: Theoretische Grundlagen

- Genauere Beschreibung von Computer Supported Cooperative Work (CSCW), Collaborative Work Space (CWS), Human Computer Interaction (HCI), Ubiquitous Computing (UbiComp), Ergonomie, Usability.

- Beispiel eines bestehenden CSCW-Systems

### **Kapitel 3: Analyse**

- Übersicht von Carola Neumanns Arbeit (Neuman (2006)) insbesondere Betrachtung der dort beschriebenen Szenarien.
- Anwendungsfälle
- Beschreibung besonderer Interaktionsschemata in einem interaktiven Gruppenarbeitsraum.
- Beschreibung einer „Multi-Desktop-Steuerung“
- Hardware- und software-technischen Rahmenbedingungen für den Prototyp

### **Kapitel 4: Prototyp - „ProjectRoom“**

- Untersuchung unterschiedlicher Sichten auf einen CWS (z. B. service-orientierte Sicht mit (teil)zentrierter Architektur versus Dezentralisiertes System, wobei jede mitgebrachte Hardware Ressourcen für das Gesamtsystem zur Verfügung stellt; thin-Client versus fat-client)
- Architektur - eine Übersicht
- Design und Realisierung
- Beschreibung des Prototyps und der zur Verfügung stehenden grafischen Benutzerschnittstellen
- Ein kleines Fazit über die technischen Aspekte des Prototyps

### **Kapitel 5: Fazit**

- Konzeptuelle Bewertung des Prototyps
- Wurden die Ziele erreicht? Wie sind die Ergebnisse zu bewerten.
- Ausblick

### **Anhänge:**

- Anhang A: Anwendungsfälle in tabellarischer Form
- Anhang B: Prototyp-Abbildungen; Gestaltung des Prototyps

## 2. Theoretische Grundlagen

Computer Supported Cooperative Work<sup>1</sup> (CSCW) ist ein interdisziplinärer Forschungsbe-  
reich und beinhaltet nicht nur die Informatik sondern auch die Soziologie<sup>2</sup> und die Organi-  
sationstheorie<sup>3</sup> - dabei haben die genannten Forschungsgebiete bezüglich CSCW ähnliche  
Zielsetzungen. Oft werden Schlagwörter respektive Akronyme wie HCI, UbiComp, CWS, Er-  
gonomie und Usability genannt und deuten schon deren Zusammenarbeit und die damit  
einhergehende Komplexität an. Nach (Peter Tandler (2004)) sind für die Entwicklung von  
Anwendungen zur synchronen computergestützten Gruppenarbeit folgende Bereiche grund-  
legend (siehe Abbildung 2.1):

1. CSCW
2. Human Computer Interaction
3. Ubiquitous computing

In den folgenden Unterkapiteln möchte ich den Kontext respektive Rahmen, in welchen sich  
diese Arbeit bewegt, näher beschreiben.

### 2.1. Computer Supported Cooperative Work

#### 2.1.1. Begriffe und Definitionen

Nachdem schon einige Male der Begriff CSCW aufgetaucht ist, hier vorerst zwei (aus einer  
Vielzahl von) Definitionen. Computer Supported Cooperative Work (CSCW):

1. bezeichnet nach Johann H. Schlichter (Schlichter und Borghoff (1998)):

---

<sup>1</sup>Deutsch: Computer- respektive Rechnergestütztes kooperatives Arbeiten

<sup>2</sup>Soziologie: Gruppenarbeit involviert Menschen, d.h. CSCW erfordert eine zusätzliche Untersuchung, die  
sich mit den Verhalten der Menschen innerhalb einer Gruppe sowie der Rollenverteilung beschäftigt - nicht  
zu vergessen, die Auswirkungen von computergestützter Gruppenarbeit auf den Einzelnen.

<sup>3</sup>Organisationstheorie: Gruppenarbeit findet meist im Kontext einer oder mehrerer Organisationen statt,  
d.h. es gibt eine Wechselwirkung zwischen Organisation und CSCW. Bei der Einführung eines CSCW-  
Systems müssen somit auch die Organisationsstrukturen berücksichtigt werden.

„... die theoretischen Grundlagen bzw. die Methodologien für Gruppenarbeit und deren Computerunterstützung.“

2. ist nach (Paul Wilson (1991))

„... a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques.“

Zu unterscheiden hiervon ist der Begriff Groupware (fälschlich synonym gebraucht), welcher ein CSCW-System meint, oder genauer:

„Ein System ist, welches Gruppenarbeit unterstützt - also eine Realisierung der im CSCW spezifizierten theoretischen Grundlagen.“

Oder, nach (Robert Johansen u. a. (1991)):

„Groupware is a generic term for specialized computer aids that are designed for the use of collaborative work groups. Typically, these groups are small project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support.“

Die Unterscheidung ist wichtig, da Groupware (oft auch als Collaborative Work Space (CWS) bezeichnet) letztendlich die Schnittmenge der in 2.1 dargestellten Bereiche ist. Da CSCW unterschiedliche Zielsetzungen hat - daher auch die unterschiedlichen Forschungsbereiche - lässt sich der Begriff dementsprechend unterschiedlich auslegen. Zum Beispiel könnte der Schwerpunkt auf den Begriff der Gruppe (Untersuchung computergestützter gruppendynamischer Prozesse) liegen; hier spräche man dann von „Rechnergestützter **Gruppenarbeit**“. Wird aber der Begriff der Kooperation ins Zentrum der Betrachtung gerückt, d. h. kooperatives Arbeiten innerhalb und auch außerhalb (Personen arbeiten planmäßig auf ein Ziel hin) der Gruppe untersucht, so wird eher von „Rechnergestütztes **kooperatives** Arbeiten gesprochen.“

### 2.1.2. Motivation für den Einsatz von CSCW

Warum gewinnt CSCW immer mehr an Bedeutung, insbesondere für Unternehmen? Was mögen die Gründe sein? Die Gründe sind u. a. in der Organisationsstruktur eines Unternehmens und in der rasanten Entwicklung neuer Technologien zu finden. Zum Beispiel treten bei Unternehmen, bedingt durch ihre hierarchische Organisationsform und teils sehr abgegrenzten Kompetenzbereichen, Kommunikationsverluste horizontaler Ebenen zugunsten vertikaler auf - im Allgemeinen führt dieses zur einer Ineffizienz in der innerbetrieblicher Koopera-

tion<sup>4</sup>. Auch stellt der Einsatz unzulänglicher Informationstechnologie in Unternehmen - z.B. unterschiedliche Informationshaltung bei diversen Mitarbeitern - ein Problem dar; die Daten mögen zwar vorhanden sein, sind aber nicht von jedem „zugreifbar“ respektive abrufbar. Zeit ist Geld, und einen Zeitverlust durch Informationsfindung kann sich ein Unternehmen beim heutigen Wettbewerb nicht mehr leisten. Seit Mitte/Ende der neunziger Jahre ist eine Vielzahl neuer Technologien, insbesondere für den Kommunikationssektor, entwickelt worden und aus der Arbeitswelt nicht mehr wegzudenken. Beginnend mit der Netzwerktechnologie<sup>5</sup> als Kommunikationsinfrastruktur, über das Internet (weltweite Vernetzung) und Intranet (lokal begrenzte Vernetzung), hin zu Anwendungen die gruppenorientierte Arbeitsprozesse ermöglichen (z. B. Videokonferenz), steht dem Anwender, privat wie auch geschäftlich, eine große Anzahl neuer Mittel zur Verfügung. Dennoch sind die verschiedenen Technologien nicht oder auch nur wenig integriert, existieren zum Teil separat, und fristen somit eine Einsiedlerdasein, spezialisiert auf bestimmte Arbeitsprozesse. So hat der Einsatz von CSCW im Wesentlichen wirtschaftliche Gründe (neben technologischen) mit der Zielsetzung, dass:

1. Produkte schneller entwickelt werden,
2. der Informationsfluss respektive -transfer schneller wird,
3. sich der Verwaltungsaufwand verringert,
4. sämtliche „Wissensträger“ untereinander „vernetzt“ sind, dass das Know-How des Unternehmens auch unternehmensweit zur Verfügung steht und somit eine bessere Ausnutzung desselben gewährleistet wird.

Natürlich darf die Organisationsstruktur eines Unternehmens nicht völlig aufgebrochen oder permanent unterlaufen werden - insbesondere auf vertikale Ebene. Dennoch können so genannte Task-Forces, speziell gebildete Teams, deren Mitglieder aus unterschiedlichen Unternehmensbereiche (Abteilungsleiter unterschiedlicher Ebenen, z. B. Finanzwesen, Produktionsleitung) stammen, temporär begrenzte Aufgaben bearbeiten - und dieses auch besser, da sich gerade das für diese spezielle Aufgabenstellung benötigte Know-How gebündelt im Team wiederfindet und sofort angewendet werden kann. Ein Unternehmen kann somit flexibler auf zukünftige Aufgaben respektive Probleme reagieren.

### 2.1.3. CS(CW) - Cooperative Work

**Work - Arbeit:** Was genau bedeutet „cooperative work“<sup>6</sup>? Beginnend mit dem englischen Begriff „work“ wird hier ein Arbeitssystem verstanden, welches, angelehnt an die Leavitt-Raute, aus den Komponenten:

<sup>4</sup>Koordinationsproblem, Informationsüberlastung, Wissen nur in einer Person gebündelt

<sup>5</sup>mehr hardware-technischer als software-technischer Art(Ethernet, WWW, u.a)

<sup>6</sup>cooperative work: Deutsch = kooperatives Arbeiten

1. Mensch
2. Aufgabe
3. Organisation
4. Technik

besteht. Sämtliche Komponenten stehen hierbei in Korrelation zu einander (Abbildung 2.2). Innerhalb des Arbeitskontextes wird unter *Aufgabe* der zu bewältigende Arbeitsauftrag verstanden. *Menschen* stehen für die Akteure, welche sich der Aufgabe annehmen - dieses abhängig von dem Kommunikationsgrad z. B. in Gruppen, alleine etc.. In und auch mit der *Organisation* findet die Aufgabenbewältigung statt, wobei die Technik hier unterstützend wirkt. Tatsächlich beschäftigen sich CSCW-Systeme hauptsächlich mit der Technik-Komponente. Für die Entwicklung von CSCW-Systemen ergeben sich dadurch spezielle Anforderungen:

1. **Mensch:** Entwurf von neuen Benutzerschnittstellen, z. B. Interaktion mit große Displays
2. **Aufgabe:** Entwurf eines Modells für mögliche parallel ablaufende Geschäftsprozesse respektive simultanes Arbeiten, z. B. a/synchrone gemeinsame Dokumentenbearbeitung.
3. **Organisation:** Entwurf neuer Regeln für die Zugriffe auf Informationen, z. B. Rechteverwaltung.
4. **Technik:** Entwurf zusammenarbeitender transparenter ubiquitäre Computersystem.

Das hier angewendete ATOM-Modell von Leavitt - eigentlich entworfen für herkömmliche Software-Systeme - findet nochmals Anwendung im Unterkapitel Software-Ergonomie, und zeigt eine andere Sicht auf die wechselwirkenden Komponenten.

**Cooperative - kooperativ:** Im Bereich der Kommunikation innerhalb einer Gruppe, stellt die Kooperation den höchsten Kommunikationsgrad dar. Eine wesentliche Rolle spielt somit die Intensität der Interaktion beteiligter Akteure. Je höher der Kommunikationsgrad, desto größer der Bedarf an Computerunterstützung. Nachfolgende Tabelle zeigt kurz die einzelnen Grade der Kommunikation innerhalb der Gruppe und deren Unterschiede (siehe Abb.: 2.3). Dabei kann der Übergang zwischen Kollaboration und Kooperation fließend sein (tatsächlich werden diese Begriffe in vielen Fällen synonym gebraucht). Bei der kooperativen Gruppenarbeit ist noch zu beachten, dass sich hier unter soziologischen Gesichtspunkten auch andere, unterschiedliche Arbeits-Modi oder Verhaltens-Modi einstellen können, wie zum Beispiel Rivalität/Konkurrenzdenken, aggressive Arbeitsweisen, lockeres koordiniertes Arbeiten, sehr enges kollaboriertes Arbeiten vice versa. Zur Kooperation ließe sich noch sagen, dass diese im Wesentlichen aus:

- Vertrauen,
- gemeinsame Ressourcen/Material,
- ausdrücklichen Koordination (gemeinsames Ziel bedingt gemeinsamen Plan = Koordination) und
- gemeinsamen Zielen

besteht. Eine andere Sichtweise auf kooperatives Arbeiten zeigt das Zusammenspiel zwischen das „gemeinsame Planen“ und der „gemeinsame Fokus“ (Schlichter und Borghoff (1998)). Je nach Höhe des Planungsgrad respektive Grad des Fokus unterscheiden sich die Interaktionen der Mitarbeiter untereinander sowie die Aufgabenbewältigung (siehe Abbildung 2.4).

#### 2.1.4. CSCW-Systeme

CSCW-Systeme lassen sich in zwei unterschiedliche Systeme einordnen:

1. Einordnung in einem Raum-Zeit-System
2. Einordnung in einem System, welches den Interaktionsgrad berücksichtigt (3K-Modell)

Die Klassifizierung nach Raum und Zeit geht auf die Arbeiten von (Johansen u. a. (1991)) zurück, wobei Gruppenmitglieder räumlich und zeitlich von einander getrennt sein können (aber nicht müssen). Dargestellt in einer Matrix ergeben sich vier mögliche Konstellationen (Abb.: 2.5). Interessant ist auch die Raum-Zeit-Matrix von (Grudin u. a. (1995)) mit der zusätzlichen Unterscheidung zwischen vorhersehbaren und nicht vorhersehbaren räumlich getrennten respektive asynchronen Arbeiten (siehe Abb.: 2.6). Diese Diplomarbeit bezieht sich, entsprechend der Raum-Zeit-Matrix, auf ein Szenario, welches das kollaborative Arbeiten zur selben Zeit und am gleichen Orte zeigt - ich werde später noch genauer darauf eingehen.

Eine andere Klassifizierung zeigt das 3K-Modell, welches die einzelnen Systeme nach deren Unterstützungsfunktion einordnet. Die Unterstützung bezieht sich hierbei auf die Intensität der Kommunikation respektive Interaktion, also:

- K1: Kommunikationsunterstützung
- K2: Koordinationsunterstützung
- K3: Kooperationsunterstützung

Natürlich gibt es auch Systeme die zwei oder drei Klassen zugehörig (mit unterschiedlichen Ausprägungen) sein können. Die 3K-Pyramide zeigt einen kleinen Überblick einzelner CSCW-Systeme im Kommunikationskontext (Abb.: 2.7). Nach (Schlichter und Borghoff (1998)) ergeben sich folgende Systemklassen (die Parenthesen bezeichnen in etwa die Unterstützungsschwerpunkte - absteigend):

1. Kommunikation (K1)
2. Gemeinsame Informationsräume (K1, K2, K3)
3. Workflow Management (K2, K1)
4. Workgroup Computing (K3, K1, K2)

Da ich mich im weiteren Verlauf dieser Arbeit auf die Systemklasse „Workgroup Computing“ beziehen werde, soll diese hier noch eingehender beschrieben werden - nähere Beschreibungen der anderen Klassen sind in (Schlichter und Borghoff (1998)) und (Wilson (1991)) zu finden.

### **Systemklasse - Workgroup Computing**

Diese Klasse zeichnet eine enge Kooperation zwischen den Gruppenmitgliedern aus. Unterstützung erfahren hier vor allen Dingen die amorphe Tätigkeiten, d.h. Tätigkeiten, welche weniger einen wiederholbaren Muster folgen - hier ist auch die Abgrenzung zu „Workflow Management“ zu sehen, bei dem die Prozessorientierung im Vordergrund steht (vordefinierte Arbeitsabläufe etc.). Auch sind die Gruppen eher kleiner gehalten und bilden meistens eine Art „Task-Force<sup>7</sup>“, welche spezielle bereichsüberschneidende Aufgaben bearbeitet. Dementsprechend müssen Workgroup Computing-Systeme eine hohe Flexibilität in Bezug auf die Aufgabeninhalte aufweisen. Einige Beispiele für Workgroup Computing Anwendungstypen wären:

- **Gruppeneditoren:** Unterstützung der gemeinsamen Dokumentenerstellung. Man unterscheidet drei Arten von Editoren:
  1. Annotationssysteme: Kommentare und Korrekturen dem Dokument befügen (Zum Beispiel als „Elektronische Notiz“)
  2. Koautorensysteme: Synchrone Dokumentenbearbeitung durch mehrere Autoren (Realzeit-Editor) oder asynchrones Bearbeiten von Dokumenten (Editor mit Benachrichtigung; z. B. mittels CVS<sup>8</sup>)
  3. Werkzeuge die eine gemeinsame Verwendung erlauben (z. B. simultanes Bearbeiten von Bildern)

---

<sup>7</sup>Task-Force = Einsatzgruppe; historisch aus der militärischen Begriffswelt entlehnt

<sup>8</sup>CVS = Concurrent Version System

- **Sitzungsunterstützungssysteme:** Auch „Electronic Meeting Systems“ (EMS) genannt. Sollen u. a. bei der Ideenfindung und -sammmlung, Strukturierung von Argumenten, Abstimmungen, Bewertungs- und Auswahlverfahren (GDSS<sup>9</sup>) unterstützend wirken.
- **Planungssysteme:** Unterstützung für die Koordination von Ressourcen und Kapazitäten

## 2.2. HCI

Wie CSCW ist auch HCI Human-Computer-Interaction (Mensch-Computer-Interaktion) ein interdisziplinärer Forschungsbereich. Die involvierten Forschungsbereiche mit ihren jeweiligen Akzenten sind:

- **Informatik** Anwendungsdesign, Entwicklung von Benutzerschnittstellen (human interfaces)
- **Psychologie** Theorien die sich mit kognitive Prozesse beschäftigen, Analyse (empirisch) von Benutzerverhalten
- **Soziologie und Anthropologie**<sup>10</sup> Interaktion zwischen Technologien, Arbeit, Organisation
- **Industrial Design**<sup>11</sup> Interaktive Produkte

Aus der Sicht der Informatik sind die letzten drei „unterstützende“ Disziplinen. HCI untersucht die Leistung gebündelter Eigenschaften von Mensch und Maschine zur Aufgabenbewältigung sowie deren Kommunikation. Dabei profitiert HCI von der Erfahrung, die in den beiden Gebieten (Mensch/Maschine) bisher gemacht wurden, und lässt diese in ihren Untersuchungen einfließen:

- **Mensch** Kommunikationstheorie, Linguistik, Sozialwissenschaften, kognitive Psychologie, Leistungsfähigkeit des Menschen, Semiotik, Grafikdesign, Industrial Design
- **Maschine** Techniken in Computergrafik, Betriebssystem, Programmiersprachen, Entwicklungsumgebungen

---

<sup>9</sup>GDSS = Group Decision Support System

<sup>10</sup>Anthropologie = Wissenschaft vom Menschen und seiner Entwicklung

<sup>11</sup>Industrial Design = Formgebung, bewusste Gestaltung von Gebrauchsgegenständen

Eine große Bedeutung kommt dem gesamten Entwicklungsprozess von Benutzerschnittstellen zu, d. h. die Analyse, die Spezifikation, der Entwurf, die Implementation und das Testen von Schnittstellen. Einen guten Überblick über die Forschungsthemen bietet Abbildung 2.8 von (Hewett u. a. (2007)). Eine Definition von HCI, welche sich mittlerweile etabliert hat, ist die von ACM<sup>12</sup> SIGCHI<sup>13</sup>:

„Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.“

### 2.2.1. Motivation und Zukunft

Es gibt implizite und explizite Einflüsse auf HCI. Letztere mögen von der rasante Entwicklung der Computerindustrie, den neuen Erkenntnissen der Informatik und die einhergehende Innovationsflut herrühren, erstere sind psychologischen und Gesellschaftspolitischen Ursachen zuzuordnen. Versucht man die Einflüsse in ihrer Art zu unterscheiden, so ergeben sich im Wesentlichen drei (korrelierende) „Bereiche“: wirtschaftliche, technische, gesellschaftliche. Beispiele hierfür wären:

- **Wirtschaftlich**

- Geringere Hardware-Kosten: Computer (nicht nur PCs) sind für jeden erschwinglich, d. h. schnellere „computerization“ der Gesellschaft und größere Verbreitung von „embedded Systems“.

- **Technisch**

- Neue Technologien erschaffen neue Einsatzgebiete: Verkleinerung der Hardware und Reduktion des Stromverbrauchs von Hardware führt zur deren Mobilität (portable) mit neuen Einsatzgebieten
- Breitband-Interaktion: Durch Erhöhung der Computer, Grafik, Netzwerkgeschwindigkeit u. a., ergeben sich neue qualitativ unterschiedliche Benutzerschnittstellen (z. B. Virtual Reality).
- Embedded Systems

- **Gesellschaftlich**

- Akzeptanz und Nutzung von Computer durch einen immer größeren Teil der Gesellschaft, führt zur einer größeren Kommunikation mittels Computertechnologie.

---

<sup>12</sup>ACM = Association for Computing Machinery, ähnlich der Gesellschaft für Informatik (GI) aber international tätig

<sup>13</sup>SIGCHI = Special Interest Group on Computer-Human Interaction

- Informationsgesellschaft (und Filterung Informationsflut)

### 2.2.2. HCI und CSCW-Systeme

Viele CSCW-Systeme benutzen neue Technologien, die, je nach Konzept des System (z. B. Grad der zu unterstützende Kommunikation, Raumausstattung), nicht nur neue Benutzerschnittstellen erfordern, sondern auch neue Interaktionsformen. Wie ist beispielsweise eine optimale Steuerung eines großen Wand-Touchscreen möglich - wobei eine Interaktion entfernt (der Anwender sitzt/steht in nicht unmittelbarer Nähe des Wand-Touchscreen) oder direkt (der Wand-Touchscreen befindet sich in Reichweite des Anwenders) erfolgen kann.

### 2.2.3. Abgrenzung - CHI, MMI

Für HCI gibt es noch zwei „alternative“ Bezeichnungen:

1. CHI „Computer-Human-Interaction“: Bezeichnete vor HCI diesen Forschungsbereich, wurde aber später durch HCI ersetzt, da hier die Betonung auf den Menschen gelegt ist - häufig noch synonym gebraucht.
2. MMI „Man-Machine-Interaction“: Im Unterschied zu HCI, steht hier nicht nur die Interaktion zwischen Mensch und Computer im Vordergrund, sondern alle Arten von Interaktion die zwischen maschinellen Teilen und Menschen erfolgen kann (z. B. Anzeigetafeln, Schalter, bei einem Kontrollpult) - HCI-Bereich kann hier als Teilmenge angesehen werden.

## 2.3. Ubiquitous Computing

Nach Mark Weiser und John Seely Brown läutet „Ubiquitous Computing“ eine neue Ära in der Computer-Anwendung ein. Waren es vormals noch viele Benutzer, die sich ein Computer-System, den Mainframe, geteilt haben, so kommt heutigen Tages auf ein Computer-System (PC) ein Benutzer. In Bezug auf die vorigen Kardinalitäten der Computer-Mensch-Relationen, strebt UbiComp eine N:1-Beziehung (oder auch N:N-Beziehung) an, d. h. viele Computer „teilen“ sich respektive unterstützen einen (oder auch viele) Menschen - Abbildung 2.9 verdeutlicht dieses. Computer sind jetzt nicht mehr Einzelplatz-Geräte, gebunden an einem festen Ort, sichtbare Hilfe für den Menschen - vielmehr durchdringen sie den menschlichen Alltag, sind überall zu finden, und verbleiben doch im Unsichtbaren. UbiComp

(Abkürzung für Ubiquitous<sup>14</sup> Computing) spricht hier von **Disappearing**<sup>15</sup> **Computer**. Mark Weiser (\* 23. Juli 1952 in Chicago, Illinois, USA, 27. April 1999), prägte als erster den Begriff des „Ubiquitous Computing“ und schrieb in einen seiner früheren Aufsätze für die „Scientific American“ (Weiser (1991)):

„[...] that the most profound technologies are those that disappear“

Die UbiComp-Forschung untersucht, wie der Alltag des Menschen durch eine ganze Sammlung von interagierenden Artefakten (interactive artifacts) unterstützt respektive verbessert werden kann und wie diese Artefakte eine neue menschenfreundliche Umgebung schaffen können - wobei der Computer eine gänzlich andere Rolle spielen wird als bisher.

### 2.3.1. UbiComp versus Virtual Reality

Während Virtual Reality<sup>16</sup> die Menschen in eine computer-generierten Welt setzt, geht UbiComp gewissermaßen den umgekehrten Weg: Computer (und Software) werden derart entworfen, dass **diese** unter den Menschen in der realen Welt *leben* können.

### 2.3.2. Themenspektrum von UbiComp

Einige Hauptthemen von UbiComp sind nachfolgend kurz beschrieben:

#### 2.3.2.1. Entwurf neuer Interaktionen

Hier wird nicht nur die Frage untersucht, wie Menschen mit *unsichtbaren* Geräten interagieren können, sondern auch der Wechsel von einer traditionellen **expliziten** Interaktion (PC-Bedienung) zur einer zukünftigen **impliziten** Interaktion.

---

<sup>14</sup>ubiquitous (engl.) = ubiquitär, allgegenwärtig

<sup>15</sup>Disappearing, disappear = verschwinden

<sup>16</sup>Virtual Reality (Abkürzung: VR), Bezeichnung für eine mittels Computer simulierte Wirklichkeit oder künstliche Welt (Cyberspace), in die Personen mithilfe technischer Geräte (elektronische Brille, Lautsprecher, Datenhandschuh u. a.) versetzt und interaktiv eingebunden werden.

### 2.3.2.2. Privatsphäre, Vertrauen und Sicherheit

Überall dort, wo Daten gesammelt werden, stellt sich die Frage nach deren Sicherheit (Zugriff, Persistenz). Die in einem ubiquitären System gesammelten und ausgewerteten Daten sind Informationen über Menschen, d.h. auch Informationen privater Art. Wie kann also die Privatsphäre innerhalb eines solchen Systems geschützt werden und wie vertrauenswürdig (Vertrauenswürdigkeit betrifft die Akzeptanz eines Systems bei dem Benutzer) ist das System für den Benutzer<sup>17</sup>? In „European Disappearing Computer Privacy Design Guidelines, Version 1.1“ von Saadi Lahlou und François Jegou (Lahlou und Jegou (2004)) sind einige Ansätze zu finden, wie dieser neuen Problematik begegnet werden kann.

### 2.3.2.3. Infrastruktur

Die auf UbiComp zugeschnittene Infrastruktur muss insbesondere zwei Eigenschaften von UbiComp-Systemen berücksichtigen:

1. Langlebigkeit: Wie könnte der Update/Upgrade dieser Systeme funktionieren, die sich gerade durch Langlebigkeit auszeichnen? Durch den potentiellen Einsatz von unzähligen Sensoren, Geräten und personalisierten Anwendungen gestaltet sich die Aktualisierung komplexer als bei herkömmlichen Systemen.
2. Robustheit: Wie reagiert das System bei Fehler und wie werden diese dem Benutzer mitgeteilt? Da das System hochverfügbar sein sollte, muss dieses u. a. auch fehlertolerant sein - nicht, dass der Ausfall respektive Fehler eines Gerätes, eine Kettenreaktion in Gang setzt und andere Geräte ausfallen lässt.

### 2.3.2.4. Kontext

Zum Kontext gehören vor allem die Parameter, die das System benutzen respektive auswerten kann, um den Menschen (teils auch andere Anwendungen) in seinen Aktivitäten zu unterstützen. Ein wichtiger relevanter Parameter mag hier der Ort (Lokation) sein, am welchen sich die Benutzer und die zum ubiquitären System gehörenden Artefakte (Hardware, Software etc.) aufhalten respektive befinden. Aus welchen zusätzlichen Parametern konstituiert sich ein bestimmter Kontext? Wie kann ein Kontext erfasst, verarbeitet und situativ als auch prophezeiend - in Bezug auf nachfolgende Aktionen des Benutzers - genutzt werden? Der Kontext wird hierbei nicht als einzelner durch Parameter beschriebener Zustand sich ablösender Prozess-Zeitpunkte gesehen, sondern ist vielmehr Teil dieses Prozesses. Betrachtet man

---

<sup>17</sup>Vertrauenswürdigkeit: Werden die Daten auch nicht weitergegeben oder anderweitig benutzt (Beispiel: Shopping Card und gezielte personenbezogene Werbung)?

beispielsweise einen Druckerdienst, der die Auswahl eines „korrekten“ Druckers davon abhängig macht, ob sich der Benutzer in der Nähe befindet (statischer Kontext), so wird schnell klar, dass bei einer Bewegung des Benutzers von einem Orte zu einem anderen (z. B. durch mehrere Büros), die einzelnen Seiten entsprechend bei unterschiedlichen Druckern zu finden sind. Wünschenswert wäre aber eine Art Vorhersage über das Ziel des Benutzers (z. B. könnte das Ziel aus den Terminkalender des Benutzers herausgelesen werden)- dann könnte der dort in der Nähe befindlichen Drucker als Zieldrucker ausgewählt werden. Hier ergäbe sich aber eine andere Schwierigkeit: Wie kann der Benutzer seinerseits den Druckort vorher sagen oder gar den Drucker selbst bestimmen? Es besteht also u. a. ein Konflikt zwischen autarke ubiquitäre Systeme und den ebenso selbständigen Benutzer, der gerne jederzeit Einfluss auf das System nehmen möchte. Eine Lösung könnte in den Entwurf neuer Interaktionstechniken liegen, die z. B. System und Benutzer in einen ständigen Dialog halten - bei dem obigen Druckerbeispiel wäre eine vorige Anfrage des Systems (Inhalt: vorgeschlagener Druckort und andere „sinnvolle“ Drucker als Auswahl) an den Benutzer denkbar.

### 2.3.2.5. Informationsfindung

Eine der Hauptanforderung eines ubiquitären Systems ist die Bereitstellung diverser Dienste, welche Informationen aus verschiedenen Quellen (z. B. Sensoren) assimilieren und filtern (z. B. nach Bedeutsamkeit, kontextbezogenen Daten) können. Die gewonnenen Informationen sollen dann vom Benutzer oder der Anwendung ausgewertet und für das Erreichen ihrer Ziele oder auch Beendigung laufender Aktivitäten eingesetzt werden.

### 2.3.3. UbiComp-Software-Infrastruktur und CSCW

Nach Tandler (2004) gibt es für eine Software-Infrastruktur in ubiquitären Umgebungen, je nach involvierten Forschungsbereich, unterschiedliche Anforderungen (siehe Tabelle 2.1). Unterschieden werden können Anforderungen,

- die UbiComp selbst betreffen,
- die UbiComp und parallel dazu HCI betreffen,
- die UbiComp speziell in einer CSCW-Umgebung betreffen.

Der letzte Punkt in der Tabelle, „Kollaboration (CSCW) mit heterogenen Geräten“, meint, dass

bei der Benutzung heterogener Geräte, die UbiComp-Software kollaboratives Arbeiten ermöglichen muss. Dieselbe Information muss auf unterschiedlichen

Forschungsbereich	Anforderung
<b>UbiComp</b>	<i>Unterstützung einer heterogenen Geräte-Landschaft</i>
	<i>Unterstützung von Multi-Computer Geräten, z.B. ein aus Segmenten (Touchscreens) bestehendes großes Anzeigegerät, bei dem die einzelnen Segmente jeweils durch ein Computer gesteuert werden.</i>
	<i>Einbeziehung von Kontextinformationen, damit ubiquitäre Geräte sich der Umwelt anpassen und entsprechende Funktionalität bereitstellen können.</i>
<b>UbiComp &amp; HCI</b>	<i>Dynamische Anpassung respektive Konfiguration von Software, da der Einsatz von UbiComp in sehr dynamischen Umgebungen stattfindet.</i>
	<i>Anpassung der Informationspräsentation: Unterschiedliche Geräte mit ihren divergierenden Anzeigemöglichkeiten sollen <b>dieselben</b> Informationen angepasst anzeigen können.</i>
	<i>Unterstützung von verteilten Benutzerschnittstellen und die in einer Interaktion beteiligten Geräte. Beispiel: Bewegen von Objekten von Desktop (PDA) zu Desktop (Wandbildschirm)</i>
<i>Interaktionen mit physikalischen Objekten: UbiComp-Software muss physikalische Objekte als Elemente einer Benutzerschnittstelle unterstützen.</i>	<b>UbiComp &amp; CSCW</b>
<i>Kollaboration mit heterogenen Geräten</i>	

Tabelle 2.1.: Anforderung an einer Software-Infrastruktur in ubiquitären Umgebungen (aus Tandler (2004))

Geräten sichtbar/lesbar und durch den dazugehörigen (neuen, unterschiedlichen) Interaktionsformen modifizierbar sein.

Die in dieser Arbeit erwähnten und beschriebenen Anforderungen sind nur ein Teil eines ganzen Anforderungskatalogs; daher sei abschließend noch auf die Arbeit von Tandler (2004) hingewiesen, in der ausführliche Beschreibungen möglicher Anforderungen für eine UbiComp-Software-Infrastruktur zu finden sind.

## 2.4. Software-Ergonomie

### 2.4.1. Ergonomie allgemein

Ergonomie<sup>18</sup> bezeichnet die Wissenschaft von der menschlichen Arbeit mit dem Ziel, die Arbeitsbedingungen den Menschen anzupassen. Ein Hauptanliegen dieser Teildisziplin der Arbeitswissenschaft ist dabei der individuelle Gesundheitsschutz, zu dem auch eine humane Arbeitsgestaltung und der Schutz der Psyche zählt. Im Wesentlichen werden hier alle Komponenten eines Arbeitsplatzes inklusive die Arbeitsabläufe (Organisation der Arbeit) betrachtet. Bei der Ergonomie werden nicht nur die technische Voraussetzungen erforscht, unter denen sich menschliche Arbeit vollzieht, sondern auch die sozialen und organisatorischen. Dabei spielt die Wirkung der Arbeit auf den Menschen eine ebenso große Rolle, wie die Erforschung der Faktoren, die zur Gestaltung menschengerechter Arbeitsplätze beitragen. Hinsichtlich des individuellen Gesundheitsschutzes beschäftigt sich die Ergonomie mit dem Kriterium der Schädigungslosigkeit (Arbeit führt nicht zu unmittelbaren Gesundheitsschäden), Ausführbarkeit (Arbeit kann physisch ausgeführt werden; z. B. Arbeitsmittel abgestimmt auf Körpermaße), Erträglichkeit (Arbeit führt auch langfristig zu keinen Schädigungen) und Beeinträchtigungsfreiheit (Psychisches Wohlbefinden; keine Über/Unterforderung, Monotonie etc.). Genauere Informationen können den DI<sup>19</sup>-Normen

- DIN EN ISO 9241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten (und auch Ergonomie der Mensch-System-Interaktion)
- DIN EN ISO 10075: Ergonomische Grundlagen bezüglich psychischer Arbeitsbelastung
- DIN EN ISO 9241-2: Arbeitsgestaltung

entnommen werden.

---

<sup>18</sup>Ergonomie: griechisch ergos = Arbeit, Werk; nomos = Gesetz

<sup>19</sup>DIN: Deutsche Industrie Norm

### 2.4.2. Software-Ergonomie

Ähnlich wie die Hardware-Ergonomie untersucht die Software-Ergonomie die Anpassungen von technischen Systemen an menschliche Belange - als technisches System steht hier die Software in Vordergrund. Die Software soll dabei den Menschen in seiner täglichen Arbeit unterstützen, hauptsächlich der geistigen Arbeit. In der Informatik wird Software-Ergonomie als Teilgebiet von HCI (siehe 2.2) angesehen und beschäftigt sich in erster Linie mit der Gestaltung von Mensch-Maschine(Computer)-Schnittstellen, sowie Benutzbarkeit und Gebrauchstauglichkeit (2.4.4) von Software. An der Entwicklung software-ergonomischer Anforderungen und Normen ist nicht nur die Informatik (beispielsweise Software-Design) beteiligt, sondern auch interdisziplinäre Forschungsgebiete wie Psychologie und die bereits erwähnte Arbeitswissenschaft. Auch hier gilt wieder, dass ergonomisch gestaltete Software die Belastung mindern soll - insbesondere die psychische Belastung, welche über Stress auch in physischen Beschwerden münden kann. Bei der Gestaltung von ergonomischer Software muss ähnlich wie bei der Leavitt-Raute

- die Art und Weise menschlicher Informationsverarbeitung
- die Aufgaben der Anwender
- das Umfeld der Organisation

berücksichtigt werden - man spricht hier auch von einer Benutzer- und Aufgabenorientierung. Bei der Gestaltung von ergonomischer Software sind zwei zentrale Felder von Bedeutung:

1. Dialog: Interaktion des Benutzers mit dem Software-System für die Bearbeitung von Aufgaben
2. Benutzeroberfläche: Präsentation (optisch, akustisch) der Information, z.B. mittels Farbgestaltung, Zeichengröße etc..

Ziel ist es, Software derart zu gestalten, dass diese leicht erlernbar, bedienbar und gut verständlich ist - auch unter Berücksichtigung der Gebrauchstauglichkeit respektive Usability (siehe 2.4.4). Die für ergonomischer Software zuständigen Gestaltungsregeln (Masken und Dialoge, Farben, Menüs, Zeichenanordnung) sind in DIN EN ISO 9241 zu finden. Für die vorliegende Arbeit ist speziell der Teil 110 (vorher 10) wichtig, welcher die „Grundsätze der Dialoggestaltung“ festlegt. Die Grundsätze habe ich in der Tabelle 2.2 zusammengefasst und kurz beschrieben.

### 2.4.3. CSCW-Systeme und Software-Ergonomie

Die zuvor genannten software-ergonomischen Kriterien an einem Software-System, sind im Wesentlichen nur für Einzelplatzsysteme konzipiert wurden. In wie weit müssen diese

<b>Grundsatz</b>	<b>Beschreibung / Beispiele</b>
Aufgabenan- gemessenheit	Unterstützung des Benutzers bei der effektiven und effizienten Erledigung von Aufgaben - Minimierung unnötiger Aktionen - Geeignete Funktionalität - Standardwerte bei Eingabefeldern
Selbstbeschreibungs- fähigkeit	Bei dem Dialog ist jederzeit erkennbar, wo man sich innerhalb dieses Dialoges befindet, welcher Dialog gerade aktiv ist oder welche Handlungen unternommen werden können und wie diese ausgeführt werden können. - Rückmeldungen, die zur Verständlichkeit beitragen - Hilfe; geeignetes Hilfe-System - Zustandsänderung des Systems
Steuerbarkeit	Steuerung des Dialogablaufs (Richtung, Geschwindigkeit) durch den Benutzer. Ein Angebot an unterschiedlichen Steuerungsarten berücksichtigt die Erfahrungen des Benutzers. - Unterschiedliche Steuerungsarten, Menüführung, Tastaturabkürzung, Mausbedienung
Erwartungs- konformität	Der Dialog muss Konsistenz aufweisen, Charakteristikum des Benutzers einbeziehen (Kenntnisse aus dem Arbeitsbereich, Ausbildung; Erfahrung) sowie allgemein anerkannte Konventionen berücksichtigen. - Konsistenz in der Belegung von Funktionstasten, Menüdarstellungsführung - Anpassung an das Benutzermodell
Fehlertoleranz	Das Arbeitsergebnis kann trotz erkennbarer fehlerhafter Eingaben erreicht werden - mit keinem oder minimalem Korrekturaufwand. - Benutzerziel ist auch bei entdeckten Fehlern zu erreichen - unerkannte Fehler bedürfen nur leichte Korrektur - Fehlerhafte Eingaben führen nicht zu einem undefinierten Systemzustand
Individualisierbarkeit	Anpassbarkeit an Benutzer und Arbeitskontext möglich - Anpassung der Menüs, Icon-Leisten etc. - Erweitern von Befehlen
Lernförderlichkeit	Anleitung des Benutzers, Unterstützung beim Erlernen des Dialogsystems - Benutzung von Metaphern; Vorgänge, Bilder, Begriffe aus dem Alltag oder Anwendungsgebiet des Dialogsystem

Tabelle 2.2.: Grundsätze der Dialoggestaltung nach DIN 9241-110

software-ergonomischen Grundannahmen für ein CSCW-System oder einer Groupware angepasst werden? Ein Hauptmerkmal von CSCW-Systeme ist die gemeinsame synchrone oder asynchrone kooperative Benutzung des Systems durch mehrere Benutzer. Die ergonomischen Prinzipien müssen also für die Koordinations-, Kommunikations- und Kooperationsunterstützung erweitert werden, so dass sie

„[...] der Spezifik des Computers als Medium gerecht werden und zugleich erkennen, dass die Kriterien der Mensch-Rechner-Interaktion zu Kriterien der rechnervermittelten Mensch-Mensch-Kommunikation erweitert werden müssen.“ [Hermann u. a. (1994)]

So müssen beispielsweise zusätzliche software-ergonomische Grundsätze gefunden werden, die die Kooperation zwischen mehreren Benutzern und den dabei auftretenden Konflikten behandeln respektive berücksichtigen. Obwohl bei den CSCW-Systemen die Gruppe in Vordergrund steht, muss auch der Einzelne die Möglichkeit haben, Anpassungen an das System vorzunehmen, die seinen Fähigkeiten, Eigenschaften und Erfahrung entsprechen. Ähnlich wie bei den Einzelplatzsystemen baut sich der Benutzer auch für ein CSCW-System ein mentales Modell auf, um Rückschlüsse und Vorhersagen über das Systemverhalten stellen zu können. Durch das mentale Modell versucht der Benutzer eine Verbindung der realen Welt mit der begrenzten Welt eines Software-Systems herzustellen. Das Modell kann zwischen einem strukturellen Modell, welches den Aufbau und den befindlichen Kontext des Systems abbildet und einem funktionalen Modell, welches Funktionen und Interaktionsmöglichkeiten dem Benutzer näher bringt, unterteilt werden. Dabei arbeitet der Benutzer mit Metaphern, die Brücken zwischen der realen Welt und den dort gemachten Erfahrung und der unsicheren und „unbekannten“ Welt des zu benutzenden Software-System schlagen sollen (der Benutzer bildet hierbei Analogien). Die Metaphern sollen somit zum Verständnis des Systems beitragen. Bei den Einzelplatzsystemen ist die „Schreibtischmetapher“ eine gängige Metapher - hier finden sich Gegenstände der realen Welt mit ihren Eigenschaften (Notizblock, Schreibtisch, etc.) in dem Software-System als „Werkzeuge“ wieder (bei Microsoft Windows-Systeme „Notepad“, „Desktop“, etc.). Wichtig in diesem Zusammenhang ist die Tatsache, dass Metaphern die reale Welt nur unscharf abbilden können und es daher auch konzeptuelle Modelle geben muss, die zielgerichtet für eine System entworfen werden und zusätzlich zur Verständlichkeit des System beitragen sollen - mit dem Nachteil des erhöhten Lernaufwandes. Bei den CSCW-Systemen gestaltet sich die einheitliche Modellbildung und Anwendung geeigneter Metaphern als schwierig, da die beteiligten Benutzer unterschiedliche Vorwissen und Erfahrungen (z. B. Kenntnisse, Erfahrungen mit/über verteilte Anwendungen) haben und dementsprechend konstruierte Modelle (insbesondere konzeptuelle Modelle) unterschiedlich gut verstehen - je größer die Anzahl der Gruppenteilnehmer je schwieriger ist es, eine für jeden verständliche Metapher oder ein leicht zu verstehendes einheitliches Modell zu finden. Zu beachten sei noch der Umstand, dass sich die Benut-

zer in einer kooperativen Umgebung befinden und auch hierfür adäquate mentale Modelle gefunden werden müssen.

Die für die Erwartungskonformität notwendige Konsistenz in CSCW-Systemen ähnelt der der Einzelplatzsystemen. Teilt man die Konsistenz in eine interne und externe Konsistenz auf, so lassen sich für CSCW-Systeme folgende Unterschiede feststellen:

1. **Interne Konsistenz:** Die Bewahrung der internen Konsistenz, die sich auf die gleichartige Gestaltung verschiedener Systemteile bezieht und dadurch u. a. den Lernaufwand und dem Umgang mit dem System erleichtern soll, muss den Umstand gerecht werden, dass unterschiedliche Benutzer unterschiedliche Lernmethoden besitzen und mit dem System unterschiedlich umgehen. Die Bewahrung oder Erhöhung der Konsistenz des Systems für den einen Benutzer, kann für den anderen Benutzer (wenn nicht zu einem inkonsistenten System, dann wenigstens) zu einem System verminderter Konsistenz führen - hier bestehen demgemäss potentielle Konflikte.
2. **Externe Konsistenz:** Die externe Konsistenz bezieht sich auf die Gleichartigkeit eines Systems mit seiner Umwelt und schließt sowohl andere Systeme als auch die Anwendungsumgebung mit ein. Der Benutzer empfindet ein System als konsistent, wenn er die in einem älteren System gewohnten Arbeitsabläufe und das dort erworbenes Wissen auf das neue System übertragen respektive anwenden kann. Es kann davon ausgegangen werden, dass die Benutzer von CSCW-Systemen unterschiedliche Arbeitsbiographien in Bezug auf die System-Anwendung, die Arbeitsabläufe und -formen haben - und es somit ein Konfliktpotential gibt.

Unterschiede zwischen Einzelplatz- und CSCW-Systeme in der Software-Ergonomie bestehen noch in den folgenden Bereichen:

- Kontext
- Kontrolle
- Grundsätze der Dialoggestaltung

#### 2.4.3.1. Kontext

Ein System lässt sich immer in einem bestimmten Kontext nutzen, und auch beurteilen. Inwieweit ein System für einen Benutzer geeignet ist, ist letztendlich abhängig vom Kenntnisstand und der Erfahrung des Benutzers. Wie bereits beschrieben, treten hier Wechselwirkungen zwischen Mensch, Computer, Organisation und der Aufgabe auf. In Bezug auf Einzelplatzsysteme kann gesagt werden, dass der Mensch als Benutzer mit seinen individuellen

perzeptiven, motorischen, kognitiven, emotionalen und sozialen Fähigkeit eine Rolle innerhalb einer Organisation verkörpert und mit Hilfe der in der Organisation eingesetzten Technik (Computer-Technik) Aufgaben bearbeitet (siehe Leavitt-Raute). Für das CSCW-System muss dieses Modell angepasst werden, da hier beispielsweise nicht nur einzelne Menschen Aufgaben bearbeiten können, sondern auch Gruppen - Abbildung 2.10 zeigt diesbezüglich eine angepasste Leavitt-Raute. Bezüglich der Aufgaben kann unterschieden werden zwischen:

- **Einzelne, gemeinsame Aufgaben:** Aufgaben können durch eine Gruppe gemeinsam bearbeitet werden, d. h. hier muss das System Möglichkeiten der Kooperation, Kommunikation und Koordination für die Gruppe bereitstellen.
- **Primäre, sekundäre Aufgaben:** Die primäre Aufgabe ist die eigentliche Aufgabe, die sekundäre bezeichnet Aufgaben, die auf der Metaebene bearbeitet werden müssen (z. B. gruppenspezifische Absprachen; wer darf was bearbeiten; Aushandlungsprozesse etc.)

Die Bearbeitung von gemeinsamen Aufgaben erhöht hierbei den sekundären Aufgabenanteil, da z. B. eine zusätzliche Kommunikation (Kooperation, Koordination) zwischen den einzelnen Gruppenmitgliedern besteht.

#### 2.4.3.2. Kontrolle

Wie bereits weiter oben beschrieben, meint „menschengerechte Arbeit“ die Arbeit, die für den Menschen ausführbar ist, ohne ihn zu schädigen oder zu beeinträchtigen und ihm Freiheiten für die Entwicklung seiner Persönlichkeit lässt - die Kontrolle bezieht sich hierbei auf die Arbeitsgestaltung. Das unter der theoretischen Arbeitspsychologie (Arbeitswissenschaft) entwickelte Kontroll-Modell besteht aus drei Komponenten, die nach ihrer Kontroll-Intensität in Stufen angeordnet ist:

1. **Stufe: *Transparenz*:** Diese Stufe bezieht sich auf die Durchschaubarkeit von Software-Systemen. Sie soll Orientierungsgrundlagen liefern und ist somit eine wichtige Voraussetzung dafür, dass sich der Benutzer eine adäquate mentale Vorstellung von dem System bilden kann.
2. **Stufe: *Antizipation, Vorhersehbarkeit*:** Ausgehend von der ersten Stufe werden hier Hypothesen und Prognosen über zukünftige Prozesse und Folgen des eigenen Handelns gebildet (Handlungsalternativen).
3. **Stufe: *Beeinflussbarkeit*:** Diese Stufe der Kontrolle ist die höchste. Nur wenn Handlungen und dessen Wirkungen abgeschätzt werden können, ist eine gezielte Einflussnahme auf Bedingungen und Prozesse der Umwelt für das Erreichen bestimmter Ziele möglich respektive sinnvoll.

Übertragen auf die Computer-gestützte Arbeit bedeutet dies, dass der Benutzer das System durchschauen und Handlungsalternativen abschätzen können muss (Stufe 1 und 2), um letztendlich das System nach seinem Willen zu beeinflussen (Stufe 3) und die Aufgabe nach seinen persönlichen Wünschen bestmöglich zu bearbeiten. Wird dieses Kontrollkonzept auf CSCW-Systeme angewendet, ergeben sich, bedingt durch den Mehrbenutzerbetrieb, auf allen drei Stufen zusätzliche Schwierigkeiten. Um die beiden ersten Stufen (Transparenz, Antizipation) im System zu gewährleisten, muss der Benutzer die Möglichkeit haben, die Aktivitäten anderer Benutzer nachzuvollziehen. Ebenso muss es dem Benutzer möglich sein, einen Einfluss auf Handlungsräume der anderen Teilnehmer auszuüben. Hier besteht natürlich ein Konfliktpotential. Auf der einen Seite möchten alle gerne ihr Wunschsystem zur optimalen Aufgabenbewältigung gestalten, auf der anderen Seite ist der Eingriff in die eigene Kontrollstruktur nicht wünschenswert, und wird zumeist abgelehnt. Abgesehen davon ergeben sich hier auch arbeitsrechtliche Schwierigkeiten, denn, um zwei Beispiele anzuführen, jeder hat ein Recht auf informationelle Selbstbestimmung und auf eine autonome Arbeitsgestaltung (freie Entfaltung der Persönlichkeit, Freiheit der Berufsausübung). Lösungen für dieses Problem bäten so genannte Aushandlungsprozesse, die ein Kompromiss zwischen hoher Selbstkontrolle und Arbeitskontrolle erreichen sollen.

#### 2.4.3.3. Dialoggestaltung

Die „Grundsätze ergonomischer Dialoggestaltung“ sind bereits in der Tabelle 2.2 beschrieben wurden. Hier möchte ich noch kurz auf den Zusammenhang zwischen dem zuvor geschilderten arbeitspsychologischen Kontrollkonzept und den software-ergonomischen Prinzipien zur Dialoggestaltung eingehen. Wie in der Abbildung 2.11 zu sehen ist, können die meisten Kriterien, einer Kontrollstufe zugeordnet werden. So trägt beispielsweise das Kriterium der „Selbstbeschreibungsfähigkeit“ zur ersten Stufe (Transparenz) Orientierungshilfen bei, in dem sie dem Benutzer genügend Informationen (z. B. Online-Hilfen) für Handlungshypothesen bereitstellt. Ähnliches lässt sich über die „Erwartungskonformität“ und der „Fehlertoleranz“ in Bezug auf die Vorhersehbarkeit sagen. Die genannten Kriterien repräsentieren die Orientierungsmöglichkeit im Kontrollkonzept, „Steuerbarkeit“ und „Individualisierbarkeit“ hingegen die Beeinflussbarkeit des Systems. Die Kriterien „Lernförderlichkeit“ und „Aufgabenangemessenheit“ werden als übergeordnete Kategorien angesehen und letzten Endes durch die anderen realisiert.

Für CSCW-System müssen die Grundsätze unter dem Hintergrund der möglichen gemeinsamen Gruppenarbeit betrachtet werden. Bearbeiten mehrere Personen eine Aufgabe, synchron oder auch asynchron, können Konflikte entstehen:

- Wird eine Selbstbeschreibungsfähigkeit vom CSCW-System gefordert, so muss be-

achtet werden, dass unterschiedliche Benutzer mit dem System arbeiten - mit unterschiedlicher Erfahrung und mentalem Modell des Systems.

- Bei der Erwartungskonformität verhält sich das ähnlich - ein Microsoft Windows-Benutzer mag eine andere Menüführung gewohnt sein, als ein Linux-Benutzer.
- Die Individualisierbarkeit in einer auch gruppenorientierten Umgebung fordert nahezu Konflikte heraus - wer bestimmt über Konfigurations- oder Standardeinstellungen im System, die alle Benutzer betreffen könnten?
- Da mehrere Personen an der Bearbeitung derselben Aufgabe beteiligt sein können, ergeben sich auch Schwierigkeiten in der Steuerbarkeit - wie gestaltet sich beispielsweise eine „Undo<sup>20</sup>-Funktion bei gemeinsam zu bearbeitenden Dokumenten in derartigen Umgebungen?
- Fordert die Fehlerrobustheit die mögliche Korrektur von eigenen Fehlern, so führt dieses bei CSCW-Systemen zu zusätzlichen Schwierigkeiten - eigene Fehler können jetzt auch bei anderen Benutzern auftauchen bzw. wirken. Wie können diese Fehler „systemweit“ korrigiert werden?

T. Hermann (Hermann u. a. (1994)) definiert für CSCW-Systeme weitere Gestaltungsgrundsätze, die sich u. a. auf das Problem der Kontrolle (wer hat welche) beziehen:

- Informationsangemessenheit
- Informationelle Moderierbarkeit
- Steuerbarkeit der wechselseitigen Beeinflussung
- Aushandelbarkeit
- Gruppenorientierte Konfigurierbarkeit
- Groupware-spezifische Fehlerrobustheit
- Groupware-spezifische Transparenz

Die Forderung nach „Informationsangemessenheit“ wird dem Umstand gerecht, dass bei einem CSCW-System größtenteils die Benutzer Informationen erzeugen oder bereitstellen - Informationen, die in Quantität und Qualität variieren können. Oftmals liegen diese Informationen unstrukturiert, undurchschaubar vor, oder es fehlen zusätzliche Kontextinformationen (siehe hierfür Neuman (2006)). T. Hermann (Hermann u. a. (1994)) schlägt daher für CSCW-Systeme zusätzliche Funktionen respektive Dialogmöglichkeiten vor, die

- es dem Benutzer erlauben, sich Kontext und Zusatzinformationen explizit anzeigen zu lassen,

---

<sup>20</sup>zurücknehmen, rückgängig machen; zurücknehmen von Eingaben

- Verweise auf Kontext und Zusatzinformation zu kreieren oder zu folgen,
- und gegebenenfalls die Darstellung zu strukturieren.

Hermann spricht in diesem Sinne auch von Ergänzbarkeit, Referenzierbarkeit und Strukturierbarkeit von Informationen respektive deren Darstellung.

Bei der „Informationellen Moderierbarkeit“ sollen zusätzliche Dialoge und Funktionen zur Verfügung stehen, die es dem Benutzer gestatten, seine Dokumente bzw. Objekte als vollständige oder informationsreduzierte Kopien (unterschiedliche Versionen) zu veröffentlichen. Der Benutzer bestimmt somit über seine Informationen und den Informationsgehalt (z. B. aus Gründen der Geheimhaltung, Datenschutz).

Der Grundsatz der „Steuerbarkeit der wechselseitigen Beeinflussung“ geht auf die Beeinflussungsmöglichkeiten der einzelnen Benutzer untereinander ein. Auf welcher Art und Weise können die Benutzer sich gegenseitig beeinflussen und wie kann diese Beeinflussung kontrolliert werden? Hermann unterteilt diese Steuerbarkeit in drei weitere:

1. **Steuerbarkeit der Erreichbarkeit:** Wie erfolgt ein Verbindungsaufbau (ist nur der Initiator aktiv oder beide) und welchen Einfluss hat der Verbindungspartner auf diesen Verbindungsaufbau? Der Benutzer soll die Möglichkeit haben, seine Erreichbarkeit einzuschränken und diese Einschränkung hinsichtlich Zeitraum, Teilnehmer und Inhalt bestimmen können. Des Weiteren soll der Benutzer Bedingungen an einem Verbindungsaufbau knüpfen können (von sehr restriktiven Verbindungsaufbau bis hin zum automatischen)
2. **Steuerbarkeit der Verteilung:** Wie können Dokumente anderen Teilnehmern zugänglich gemacht werden und wie kann man den Erhalt von Dokumenten beeinflussen? Der Benutzer soll hier wählen können, zwischen aktiven Versand oder passiven Zugriff auf Dokumente.
3. **Steuerbarkeit des Zugriffs:** Wie kann der gemeinsame und individuelle Speicherbereich (z. B. Dokumentenverzeichnisse) von einander abgegrenzt und den Teilnehmern differenziert zur Verfügung gestellt werden? Dieses betrifft insbesondere die gemeinsame Bearbeitung von Dokumenten (z. B. Konflikte bei paralleler Bearbeitung gleicher Textstellen).

Beim Grundsatz der „Aushandelbarkeit“ sollen Lösungsvorschläge bei Interessenkonflikte angeboten werden. Das heißt u. a., dass Benutzer, die von Aktionen anderer Benutzer betroffen sind, hier die Möglichkeit der Beeinflussung haben (beispielsweise in Form einer Zustimmung, Ablehnung oder Modifizierung der Aktion). Die „Aushandelbarkeit“ bezieht sich unmittelbar auf die zuvor erwähnten Prinzipien, deren Rahmen im CSCW-System durch die beteiligten Benutzer „ausgehandelt“ werden müssen - z. B. die Art und Weise des Informationsaustausches.

Bei der „Gruppenorientierte Konfigurierbarkeit“ können Gruppen ihre eigene Systemkonfiguration festlegen oder wählen.

Die „Groupware-spezifische Fehlerrobustheit“ bezieht sich auf Fehler, die sich „systemweit“ und somit auf anderen Benutzer auswirken können. Derartige Fehler sollen im System entweder nicht möglich oder „systemweit“ zu beheben respektive zu behandeln sein.

Besteht eine Unklarheit der Benutzer untereinander über den Bearbeitungszustand von gemeinsamen Prozessen oder der Erreichbarkeit der Benutzer im System, so müssen hierüber neue Informationen generiert und den einzelnen Benutzern zur Verfügung gestellt werden. Das Prinzip der Transparenz beschäftigt sich u. a. mit dieser Thematik. Der Begriff der Transparenz wird in der Literatur unterschiedliche verstanden. Für Hermann ist ein CSCW-System nach den Grundsätzen der Transparenz gestaltet, wenn

- Daten zur möglichen und zur aktuellen Nutzung von Funktionen [...] abgespeichert werden können.
- Die abgespeicherten Transparenzdatensätze bei Bedarf von Teilnehmern abgerufen werden können und entsprechend ihren Informationsbedürfnissen in unterschiedlicher Detaillierung auf einem Ausgangsmedium dargestellt werden können (aber nicht müssen). Hierzu gehört auch, dass der Nutzer Voreinstellungen wählen kann, aufgrund derer ihm zeit- oder situationsabhängig bestimmte Transparenzdatensätze automatisch angezeigt werden

Nach H. Kahler („Von der Empirie zur Gestaltungsanforderung.“ in (Hermann u. a. (1994))) hingegen

„dient die Transparenz dazu, die von dem System bereitgestellte Funktionalität und deren Nutzung für den Systemnutzer sichtbar und nachvollziehbar zu machen, wie sie dies wünschen“.

Kahler unterscheidet zwischen funktionaler (Selbstbeschreibungsfähigkeit des Systems) und nutzungsbezogener (Informationen über die Systemnutzung anderer Benutzer) Transparenz und bezieht die Kommunikation, genauer die Kooperation unter den beteiligten Benutzer, enger mit ein.

Die erweiterten Grundsätze der Dialoggestaltung für CSCW-Systeme haben zum Teil gegensätzliche Ziele. Zum Beispiel werden bei der „Informationsangemessenheit“ und „Transparenz“ neue teil-strukturierte respektive zusätzliche Informationen (beispielsweise Informationen für die sekundäre Arbeiten) erzeugt, gesammelt und zur Verfügung gestellt, während die „Informationelle Moderierbarkeit“ Informationen beschränken möchte und somit eher einen informationsreduzierenden Charakter besitzt. Deshalb kommt auch dem Prinzip der „Aushandelbarkeit“ eine besondere Rolle zu, welches mögliche Konflikte unter den anderen Prinzipien zu mindern versucht oder „vertraglich“ zu regeln.

#### 2.4.4. Usability

Mit Usability ist die Gebrauchstauglichkeit oder auch Brauchbarkeit gemeint. Der Begriff selbst ist eine Zusammensetzung aus den englischen Wörtern *to use* und *ability*, zu Deutsch *benutzen* und *Fähigkeit*. Der Maßstab der Gebrauchstauglichkeit setzt sich zusammen aus:

- **Effektivität:** Erreichen des Arbeitsziels
- **Effizienz:** Aufwand für das Erreichen des Arbeitsziels
- **Zufriedenstellung:** Benutzersicht - Mängel, Beeinträchtigung der Benutzung

Wie können also bestimmte Ziele effektiv, effizient und zufrieden stellend unter Berücksichtigung eines bestimmten Nutzungskontext mit einem Produkt erreicht werden. Der Nutzungskontext selbst beinhaltet den Benutzer, die Arbeitsaufgabe, die Arbeitsmittel und die Umgebung. Usability bezieht sich nicht ausschließlich auf die Entwicklung von Software-Produkten, sondern auch auf ganz alltäglichen Gebrauchsgegenständen wie z. B. Becher, Messer etc.. Besonders interessant ist der Grad der Zufriedenstellung, der eine subjektive Komponente in die Bewertung mit einbringt. Beispielsweise mag die Teekanne von Donald A. Norman (Norman (2002)) effektiv und in gewissen Grenzen effizient sein, aber bestimmt nicht zufriedenstellend (siehe Abb. 2.12). Weitere Angaben respektive Informationen über Usability sind in der **ISO Norm 9241** zu finden.

### 2.5. CSCW-System - Collaborative Workspace - Ein Beispiel

Als Beispiel möchte ich hier das Roomware-Projekt des Fraunhofer Institut anführen, dessen Entwickler-Team Erfahrungen in den genannten Forschungsbereichen gemacht und diese in unterschiedlichen Komponenten von Roomware einfließen lassen haben. Roomware bedient sich der BEACH-Software-Plattform. BEACH steht für „Basic Environment for Active Collaboration with Hypermedia“ und ist eine Software-Architektur und Framework für Roomware-Anwendungen - sie ermöglicht die Bearbeitung von gemeinsam genutzten Daten durch mehrere Benutzer. Ursprünglich hat sich Roomware aus dem i-LAND-Projekt heraus entwickelt und stellte Basisfunktionalität für die Zusammenarbeit von Projektteams zur Verfügung - erst später entstand ein „Framework für kooperative Software“. Das i-LAND-Projekt stellt - im wahrsten Sinne des Wortes - den Überbau für Roomware (und andere Projekte) dar und hat

als Leitbild das „kooperative Gebäude<sup>21</sup>“. Kommunikation und Kooperation wird in dem gesamten Gebäude zur Verfügung gestellt. Es agiert und reagiert auf neue/veränderte Situationen und stellt kontextbezogene Informationen mittels Einsatz von aktiven und aufmerksamen Komponenten bereit (siehe IPIS (2007)).

### 2.5.1. Roomware - Ein Überblick

Einen guten konzeptionellen Überblick für eine Umgebung eines CSCW-Systems bietet die Abbildung 2.14 die Roman (Bartnik (2006)) für seine Diplomarbeit „Weiterentwicklung einer Technologiebasis für interaktive Gruppenarbeitsräume“ erstellt hat, die u. a. Roomware als Vorbild hatte. Im Projekt-Raum stehen den Benutzern eine Reihe von Ein- und Ausgabegegeräten zur Verfügung, die alle miteinander vernetzt sind und mit den gleichen Datenbestand arbeiten. Roomware setzt folgende Hardware-Komponenten ein (Abb. 2.13):

- **ViewPort:** Personal Digital Assistant (PDA)
- **CommChair:** Mobiler Stuhl mit Computer und berührungssensitives Display; autonome Stromversorgung; WLAN-Anbindung
- **ConnectTable:** Computer-Stehpult mit Display; spontane Verbindung mit einem zweiten ConnectTable möglich (UbiComp)
- **DynaWall:** Großes interaktives Display (4,50m breit, 1,10m hoch), bestehend aus drei berührungssensitiven Displays. Jede Ein-Ausgabe-Einheit hat einen eigenen PC als Kontrolleinheit.
- **InteractTable:** Mobiler Tisch mit eingelassen Touchscreen<sup>22</sup>; gedacht für kleinere Gruppen bis zu sechs Leute.

Nach (Tandler (2004)) wurde der Raum u. a. für synchrone Zusammenarbeit in ubiquitären Computer-Umgebung konzipiert - in der Raum-Zeit-Matrix (Abb.: 2.5) erster Quadrant, links oben.

### 2.5.2. Roomware - UbiComp

Da der Raum speziell auf UbiComp ausgelegt ist, sind dementsprechend zahlreiche Beispiele zu finden. Zwei davon wären:

---

<sup>21</sup>i-LAND spricht hier vom „Cooperative Building“

<sup>22</sup>Touchscreen = berührungssensitiver Bildschirm/Display

1. **DynaWall:** Da drei Segmente eine große homogene interaktive Fläche ergeben sollen, muss die Bildschirmausgabe der drei verantwortlichen PCs synchronisiert werden. Objekte, die ein Segment verlassen, müssen gegebenenfalls in einem angrenzenden Segment auftauchen respektive dargestellt werden - dieses natürlich ohne Zeitverzögerung. Eine der Anforderungen von UbiComp ist die Unterstützung von Geräten, die aus mehreren eingebundenen Computer bestehen (engl.: Multiple-Computer Devices) sowie die Koordination und Kupplung der auf den Computer laufenden Software.
2. **CommChair:** Da der CommChair auch kollaborierend mit z. B. der DynaWall eingesetzt werden kann, müssen die unterschiedlichen Anzeige-Flächen berücksichtigt werden - UbiComp-Software muss die Anzeige und Bearbeitung der gleichen Daten bei heterogenen Geräten gewährleisten (Anpassung der Ausgabe und der Interaktionskomponenten).

### 2.5.3. Roomware - HCI

Bei den in Roomware eingesetzten teils neuen Technologien, müssen bewährte Benutzerschnittstellen verbessert bzw. angepasst werden oder gänzlich neue entwickelt werden. Speziell für UbiComp-Software muss eine Trennung der Benutzerschnittstelle und der Funktionalität der Anwendung erfolgen. Auch hier zwei Beispiele:

1. **InteracTable:** Anders als bei vertikalen Anzeigegeräte, die eine definierte Ausrichtung haben (oben, unten, links, rechts), hat dieses Ein- und Ausgabemedium keine vordefinierte Orientierung - die Anwender dürfen sich wahllos um den Tisch verteilen.
2. **DynaWall:** Da die DynaWall recht groß ist, muss es eine Möglichkeit für den davor stehenden Benutzer geben, entfernte Objekte zu erreichen oder Objekte in nicht erreichbaren Regionen zu Positionieren, ohne die eigene Position verändern zu müssen (ein Lösung wäre Drag-And-Throw Abb. 1.1).

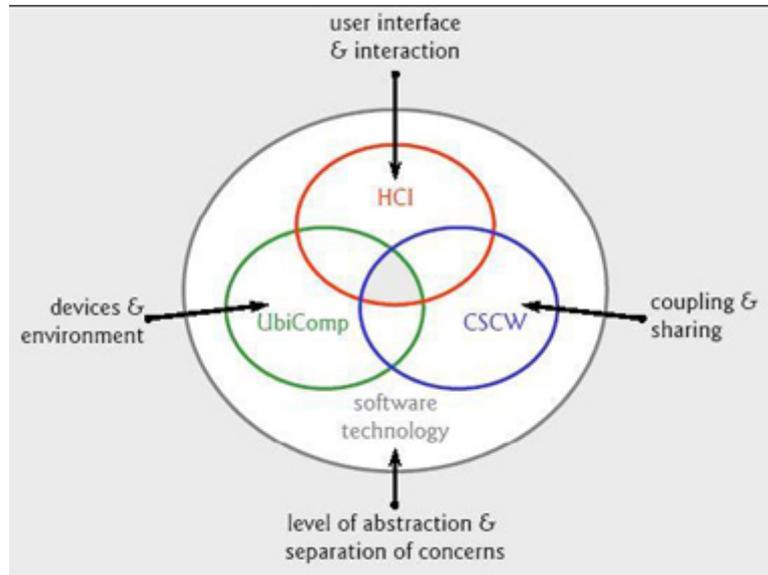


Abbildung 2.1.: Einzelne Forschungsbereiche

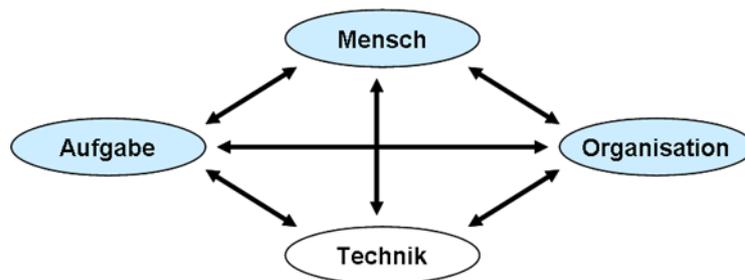


Abbildung 2.2.: Leavitt-Raute: Wechselwirkungen zwischen den Komponenten: Mensch, Organisation, Arbeit, Technik - CSCW-System beziehen sich vorwiegend auf die Technik-Komponente



Time/Space-Matrix		Zeit	
		Gleich (synchron)	Verschieden (asynchron)
Ort	Gleich (zentral, lokal)	Gemeinsames Meeting, Präsentationen	Projektmanagement
	Verschieden (dezentral, global)	Mehrautoren-systeme, Videokonferenzen	E-Mail, Workflow-Systeme

Abbildung 2.5.: Raum-Zeit-Matrix: Mögliche Anwendungen (Groupware), abhängig von Raum und Zeit

Jonathan Grudin Time/Space-Matrix		Zeit		
		gleich	Verschieden vorhersehbar	Verschieden nicht vorhersehbar
Ort	gleich	Gemeinsames Meeting, Präsentationen	Schichtarbeit	Team Rooms, Projektmanagement
	Verschieden vorhersehbar	Video-/Tele-Konferenz	E-Mail	Kollaboratives Verfassen von Dokumenten
	Verschieden nicht vorhersehbar	Interaktive Multicast Seminare, Electronic Conferencing	Computer Boards	Vorgangsbearbeitung

Abbildung 2.6.: Raum-Zeit-Matrix nach (?): Zusätzliche Unterteilung in Vorhersehbarkeit bei getrennt räumliches/zeitliches gemeinsames Arbeiten

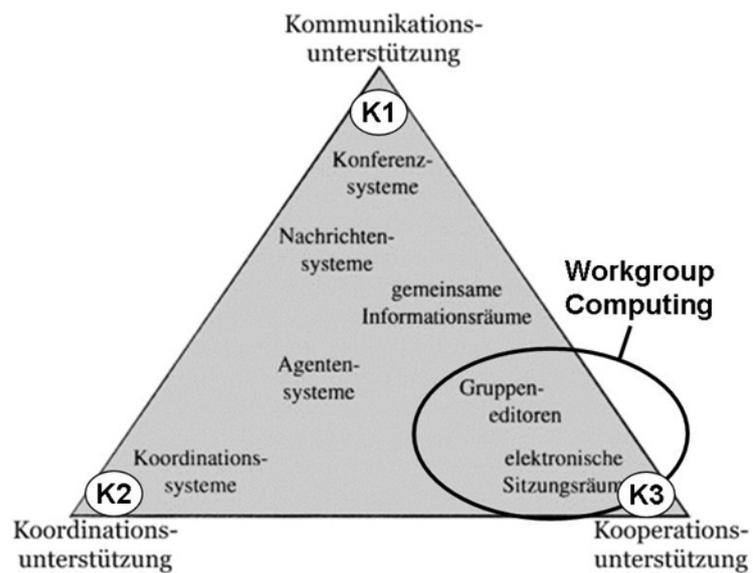


Abbildung 2.7.: 3K-Pyramide - Kommunikations-, Koordinations-, Kooperationsunterstützung einzelner Systeme

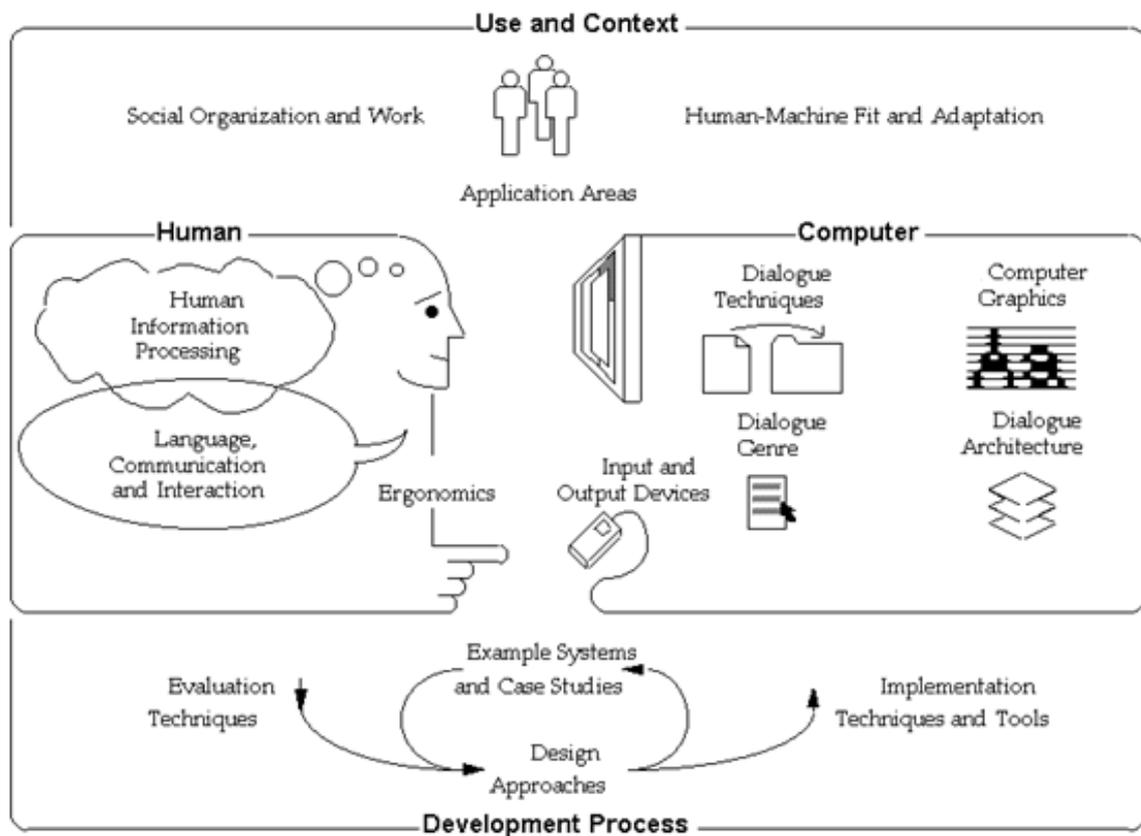


Abbildung 2.8.: Forschungsthemen und dessen Beziehung zueinander

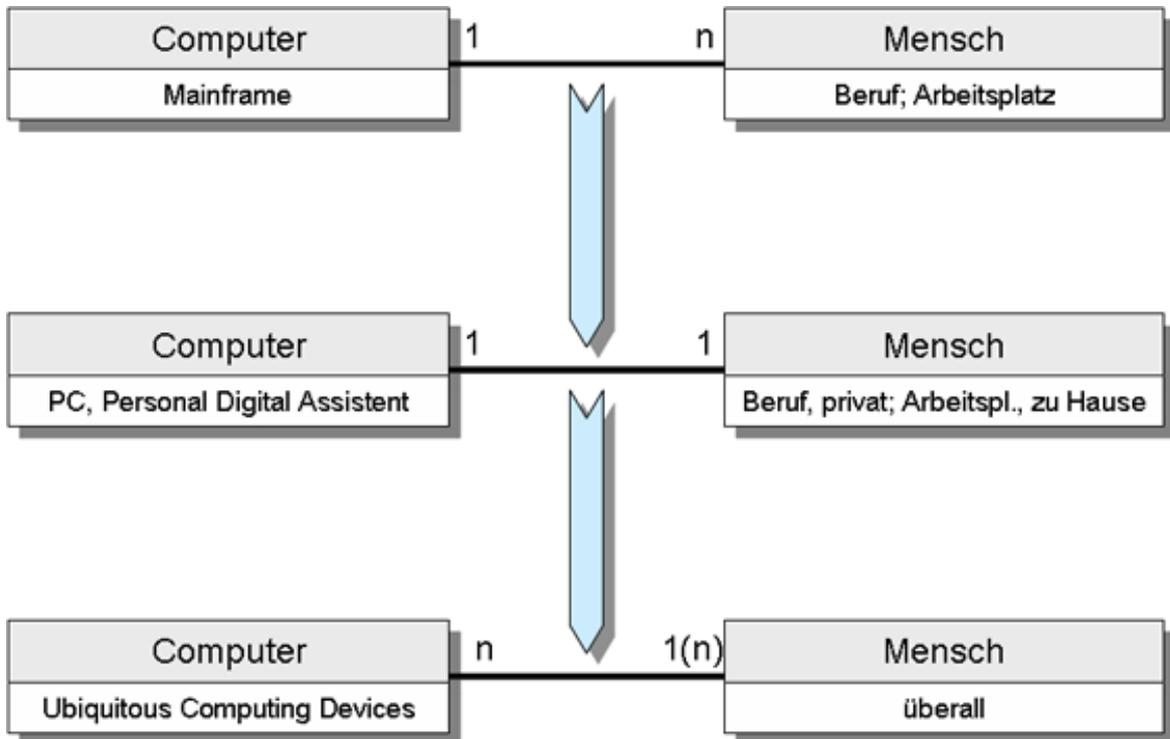


Abbildung 2.9.: Haupt-Trends in Computing

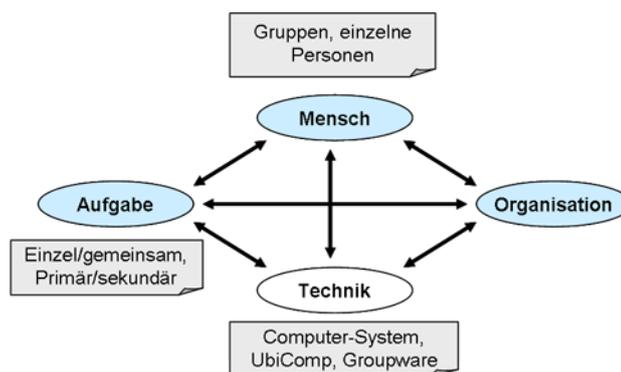


Abbildung 2.10.: Abgewandelte Leavitt-Raute im CSCW-Kontext

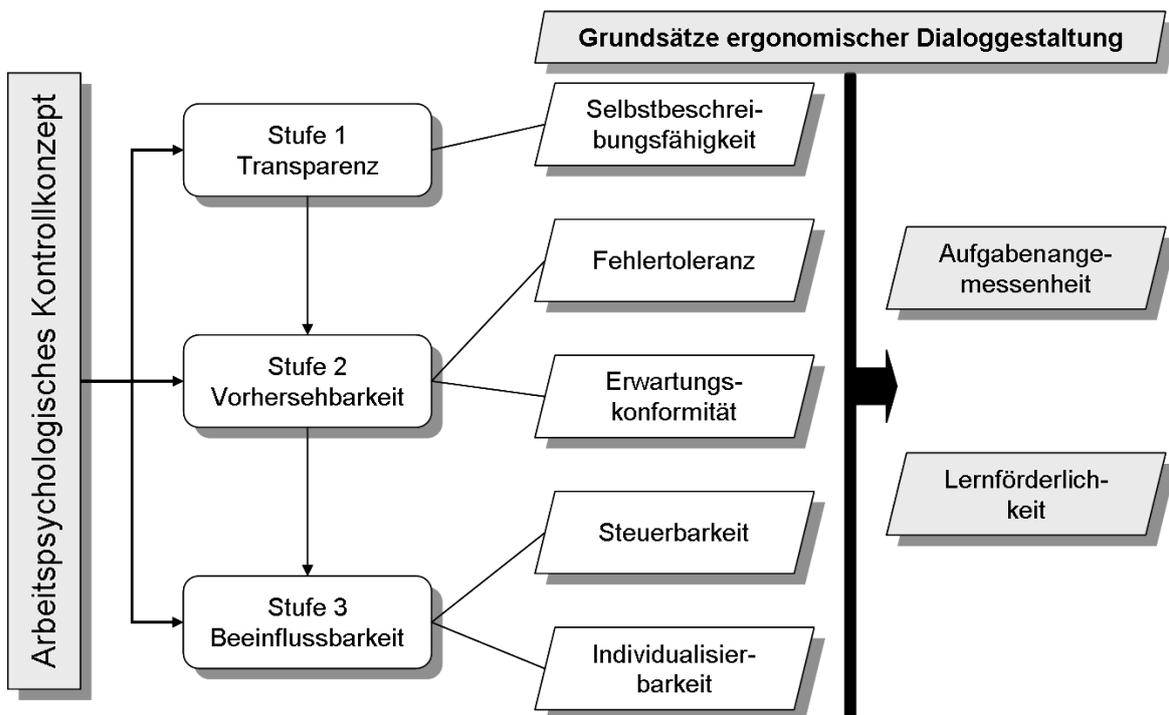


Abbildung 2.11.: Zusammenhang zwischen Arbeitspsychologischen Kontrollkonzept und den Grundsätzen ergonomischer Dialoggestaltung



Abbildung 2.12.: Usability - Design einer Teekanne (aus „The Design of Everyday Things“ von Norman (2002))

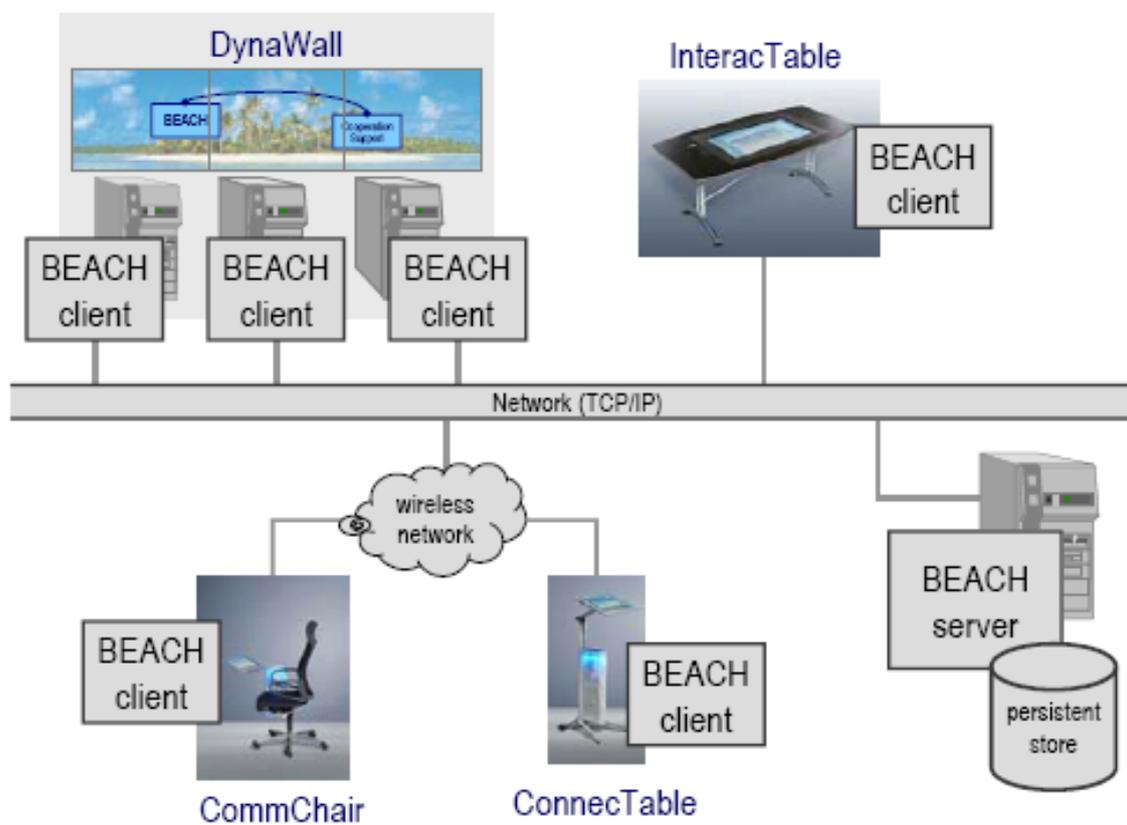


Abbildung 2.13.: Zeigt unter Berücksichtigung der BEACH-Architektur die Roomware-Komponenten

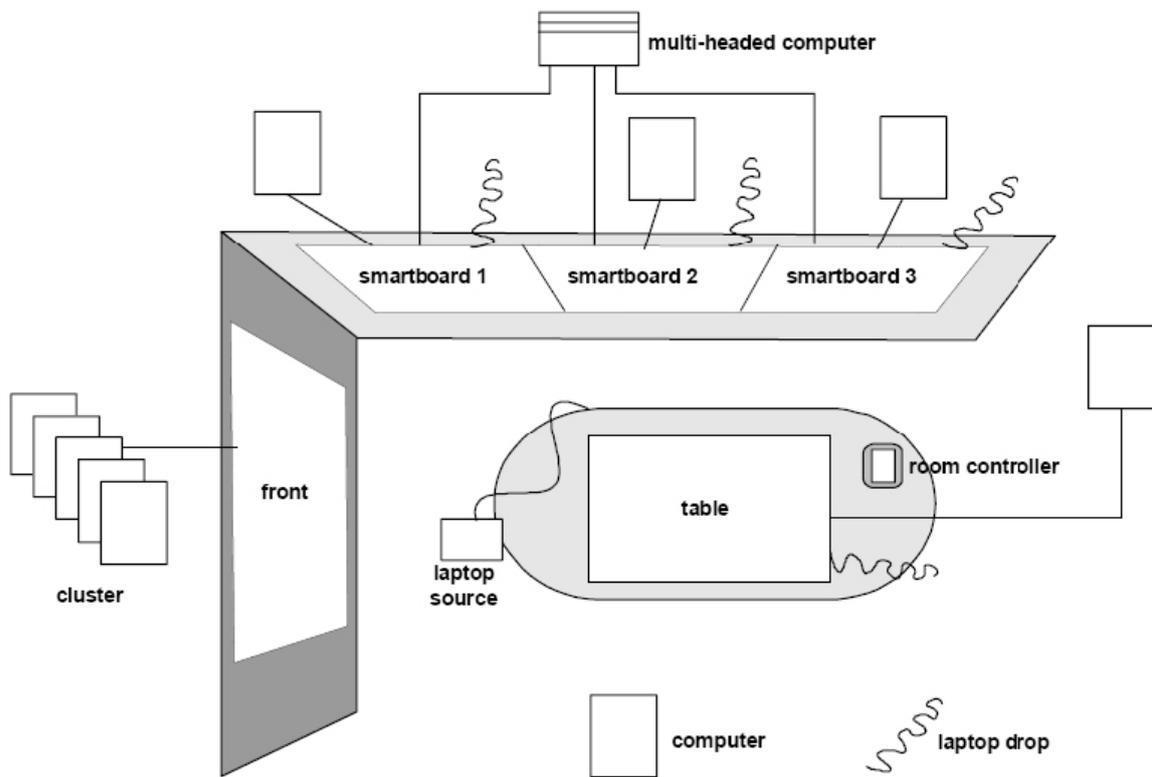


Abbildung 2.14.: Roomware-Projekt: Überblick

## 3. Analyse

Wie im Kapitel 1.1 „Zielsetzung, Methodik und Abgrenzung“ bereits erwähnt, beschäftigt sich diese Arbeit mit ausgewählten Interaktionen in einem CSCW-System und versucht diese auf bestimmte Szenarien anzuwenden. Hauptaugenmerk wird hierbei auf eine Kernfunktionalität gelegt, welche ein CSCW-System dem Anwender zur Verfügung stellen muss, um ein minimales Arbeiten mit ebendiesem zu gewährleisten. Die Realisierung, in Form eines Prototyps, wird im Kapitel ?? unter software-ergonomischen Kriterien für CSCW-Systeme und -Anwendungen sowie unter funktionalen Gesichtspunkten untersucht und ausgewertet. Im vorigem Kapitel sind die zum CSCW gehörenden und für den weiteren Verlauf der Arbeit wichtigen theoretischen Grundlagen aufgezeigt sowie ein konkretes Beispiel eines bestehenden CSCW-System gegeben worden. Ein für den weiteren Verlauf dieser Arbeit wichtiges Kapitel war das der Software-Ergonomie (Kapitel 2.4.2), welches insbesondere auf software-ergonomische Aspekte bei der Gestaltung von CSCW-Systemen eingegangen ist und erste Ansätze für die spätere Bewertung der Interaktionen und Funktionalität in einem CSCW-System bietet.

Diese Arbeit bezieht sich u. a. auf zwei Diplomarbeiten, welche sich mit dem Thema „Computer Supported Cooperative Work“ auseinander gesetzt haben - beide mit unterschiedlichen Themenschwerpunkten. Die Arbeiten sind aus dem KORA-Projekt<sup>1</sup> der Hochschule für angewandte Wissenschaften (HAW) Hamburg hervorgegangen:

1. *„Weiterentwicklung einer Technologiebasis für interaktive Gruppenarbeitsräume“* von Roman Bartnik (Bartnik (2006)): Diese Arbeit beschäftigt sich mit dem Entwurf einer neuen Architektur eines CSCW-System (spezieller: einer Infrastruktur) für den Einsatz in gruppenorientierten Projekträumen - ähnlich der BEACH-Architektur von (Tandler (2002)).
2. *„Effizienzsteigerung von Diskussionsprozessen in einem neu gestalteten Konferenzraum“* von Carola Neumann (Neuman (2006)): Carola Neuman hingegen, legt den Themenschwerpunkt auf eine Einsatzmöglichkeit in einem derart gestalten interaktiven Gruppenarbeitsraum. Als Ausgangsszenario dient ihr hierbei der Software-Entwicklungsprozess, dessen einzelnen Stufen (Voruntersuchung, Analyse, Design, etc.) durch das CSCW-System unterstützt werden sollen.

---

<sup>1</sup>Das Kora-Projekt beschäftigt sich mit der Entwicklung eines Konferenzraumes, der diskussionsunterstützende Techniken in Form von Hardware und Software enthalten soll.

Dabei wird auf die in beiden Arbeiten beschriebenen CSWC-Szenarien Bezug genommen, welche sich in erster Linie durch eine allgemeinen, generellen (Bartnik: Konferenzsystem für unterschiedliche Projekte) und einen spezielleren fachlichen Anwendungsbereich (Neumann: Szenario eines Software-Entwicklungs-Prozesses) unterscheiden. Gegenstand dieser Arbeit ist die Betrachtung von Interaktionen zwischen Mensch und CSCW-System. Untersucht werden typische Anwendungsfälle, die sich bei einer Benutzung eines gruppenorientierten Arbeitsraumes ergeben. Die aus den Anwendungsfällen resultierten später in einem Prototyp realisierten Funktionen und den damit verbundenen Interaktionen, sollen unter software-ergonomischen Gesichtspunkten untersucht und bewertet werden.

In diesem Kapitel wird zuvorderst auf die Rahmenbedingungen eingegangen, die sich sowohl auf hardware- und software-technischen Komponenten beziehen als auch soziologischen. Alsdann erfolgt die Beschreibung von möglichen Szenarien, die dann durch konkrete Anwendungsfälle ausgefüllt werden. Die in den Anwendungsfällen untersuchten Interaktionen und der daraus folgende erste Entwurf einer „Steuerung entfernter Desktops“ schließt dieses Kapitel ab.

### 3.1. Rahmenbedingungen und Abgrenzung

Da CSCW mittlerweile einen großen Forschungsbereich in der Informatik darstellt, mit unterschiedlichen Themenschwerpunkten (HCI, UbiComp, Software-Ergonomie, Usability) und zusätzlich beteiligten interdisziplinären Forschungsgebieten (Soziologie, Organisationstheorie etc.), müssen vorher die Grenzen und Rahmenbedingungen dieser Untersuchung aufgezeigt werden. Detaillierte den einzelnen Disziplinen untergeordnete Untersuchungen sind nicht möglich und würden den Rahmen dieser Arbeit sprengen. Folgende Fragen gilt es daher zu beantworten:

- Wo, in welcher **Umgebung** findet die Untersuchung statt?
- Wie, mit welchen **Mitteln** wird die Untersuchung durchgeführt?

#### 3.1.1. Umgebung

Mit der Umgebung ist einerseits die für ein CSCW-System notwendige hard- und software-technische Ausstattung gemeint, andererseits der soziologische Kontext (Rollenzugehörigkeit, Situationen) in der sich der Gegenstand der Arbeit bewegt.

Die Rahmenbedingungen für die hardware-technische Ausstattung liefert der CSCW-Raum der HAW-Hamburg (siehe Abbildung 3.1). Dieser besteht aus:

- Powerwall, frei in ihrer Form konfigurierbar

- Touchscreen
- Tabletop
- Festen PC-Arbeitsplätzen oder mobilen (z. B. Laptops)
- Wireless Lan Access



Abbildung 3.1.: CSCW-Raum in der HAW-Hamburg

Alle Komponenten sind miteinander vernetzt und haben als software-technische Rahmenbedingung Microsoft Windows XP oder ein aktuelles Linux-System als Betriebssystem installiert. Hardware-technische Restriktionen gibt es hinsichtlich der Auflösung der Ausgabe-Systeme (Powerwall, Laptops, Interactable) und der Geschwindigkeit des Wireless-Lans - die Auflösung sollte mindestens 640x480 Pixel, die Geschwindigkeit 54 Mbps<sup>2</sup> nach „IEEE 802.11h<sup>3</sup>“ betragen. Für den späteren Interaktionstest sollte jeder Teilnehmer mindestens ein Gerät im Raum erhalten, respektive sein eigenes Gerät (z. B. Laptop) in den Raum einbringen können (UbiComp). Einen guten Überblick bietet die Abbildung 2.14 (S.52) von Roman Bartnik, die auch den CSCW-Raum der HAW sehr gut skizziert. Bei der Beschreibung

<sup>2</sup>Mbps = engl. Megabit per second, Einheit der Datentransferrate

<sup>3</sup>IEEE = Institute of Electrical and Electronics Engineers, bildet Gremien für die Normung von Techniken, Hardware und Software; 802.11h = Norm für drahtlose Netzwerkkommunikation

des Prototyps, Kapitel 4, wird noch genauer auf die Hardware und Software eingegangen. Der soziologische Kontext wird durch die gruppenspezifische Nutzung des Raumes beschrieben. Mindestens zwei Benutzer nehmen an einem gruppenorientierten Diskussionsprozess teil und werden dabei durch die zur Verfügung stehenden Hilfsmittel (Hardware, Prototyp) unterstützt.

### 3.1.2. Mittel

Die aus der Analyse hervorgehenden Interaktionen, werden in einem Prototyp realisiert. Die in Kapitel 4: „Prototyp“ beschriebenen software-technischen Entwicklungskomponenten (Architektur, Design und Realisierung) sind nicht Hauptgegenstand dieser Arbeit. Der Prototyp soll die Interaktionen „simulieren“, mit einer ausreichend großen Realitätsnähe. Eine Bewertung der Architektur findet nur in einem kleinen Rahmen statt, die für die Interaktion notwendigen visuellen Gestaltungselemente hingegen werden genauer betrachtet.

### 3.1.3. Grenzen der Bewertung

Die Bewertung der durch den Prototypen realisierten Interaktionen erfolgt hauptsächlich unter software-ergonomischen Kriterien für CSCW-Systeme (siehe Unterkapitel 2.4.3). Sicherheitsrelevante Themen wie Authentifizierung, Rechteverwaltung, Rechtestrukturen und Datenhaltung in einem CSCW-System werden nicht detailliert behandelt - es sei denn Interaktionen sind unmittelbar betroffen (z. B. Systemanmeldung, Login).

## 3.2. Szenarien

Neumann geht u. a. der Frage nach, inwiefern der Software-Entwicklungsprozess durch einen CSCW-System („neugestalteter Konferenzraum“) unterstützt werden kann. Als zu untersuchendes Software-Entwicklungsmodell wird ein iteratives Modell gewählt, bei dem die einzelnen definierten Projektphasen mehrmals durchlaufen werden - geänderte Anforderungen und zusätzliche Erkenntnisse können somit in das Projekt einfließen. Konkret untersucht Neuman die Szenarien, die aus dem WAM<sup>4</sup>-Ansatz hervorgehen. Der WAM-Ansatz

„...[] ist selbst kein definiertes Vorgehensmodell, sondern liefert einen Metaprozess für die Softwareentwicklung, welcher eine evolutionäre Vorgehensweise mit Prototyping beinhaltet. Ein zentraler Punkt des WAM-Ansatzes ist der Autor-Kritiker-Zyklus.“ (Neuman, 2006, nach)

---

<sup>4</sup>WAM-Ansatz: Werkzeug & Material-Ansatz

Dabei wird der Software-Entwickler als Autor, der eigentliche Benutzer als Kritiker verstanden. Der Autor-Kritiker-Zyklus (siehe Abbildung 3.2) wird hierbei als permanenter Wechsel zwischen Analysieren, Modellieren und Bewerten angesehen. Durch die enge Zusammenarbeit zwischen den Software-Entwicklern und den späteren Benutzern soll eine möglichst anwenderfreundliche Software entstehen.

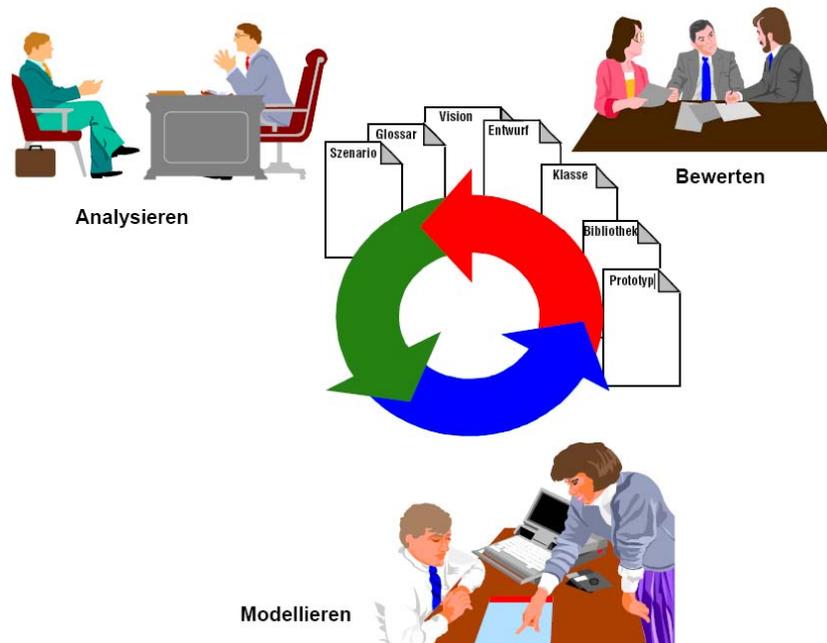


Abbildung 3.2.: Autor-Kritiker-Zyklus beim WAM-Ansatz, (nach Neuman, 2006, S. 10)

### 3.2.1. Szenarien eines Software-Entwicklungsprozesses

Die Szenarien gehen, so Neumann, aus typischen und für das Projekt wesentlichen Aufgaben hervor. Exemplarisch geht Neuman auf folgende Aufgabensituationen ein und beschreibt hierfür mögliche Szenarien:

- **Aufwandsabschätzung und Angebotserstellung**
  - Szenario 1:* Einblick in die Arbeitsabläufe beim Kunden
  - Szenario 2:* Entwurf einer Architektur
- **Fein-Analyse**
  - Szenario 3:* Präzise Zeit- und Ressourcenplanung
  - Szenario 4:* Verfeinerung der Use-Cases<sup>5</sup>

<sup>5</sup>Use-Case: engl. Anwendungsfälle

- **Projektplanung und Entwurf**
  - Szenario 5:* Speicherung von Testdaten
  - Szenario 6:* Einigung über Technologie
- **Umsetzung und Implementierung**
  - Szenario 7:* Präsentation eines Prototyps
  - Szenario 8:* Zwischenstand
- **Auslieferung**
  - Szenario 9:* Vorbereitung für Schulung und Einführung
  - Szenario 10:* Abschlusspräsentation
- **Krisensituationen**
  - Szenario 11:* Drohender Verzug

Des Weiteren beschreibt Neumann die bei einem Software-Projekt vorkommenden unterschiedlichen Rollen, welche unterschiedliche Anforderungen an die Projektdaten sowie Sichtweisen auf das Projekt haben. Mögliche Rollen wären:

- **Systemhaus:** Vorstand, Abteilungsleiter, Projektleiter, Architekt, Entwickler
- **Kunde:** Vorstand, Abteilungsleiter, Organisator, Benutzer
- **Lieferant:** „Vertriebler“, System-Spezialist

Die detaillierten Beschreibungen der einzelnen Szenarien und Rollen sind (in Neuman, 2006, S. 12 - 25) zu finden. Ziel war es, die Gemeinsamkeiten der unterschiedlichen Situationen und Szenarien herauszuarbeiten, erste Anforderungen zu definieren und ein Benutzungsmodell für die einfache und intuitive Bedienbarkeit des Konferenzsystems zu entwickeln. Erste Gemeinsamkeiten ließen sich bei den Workflows<sup>6</sup> feststellen, welche den einzelnen Situationen zugeordnet waren. So ist in den aufgeführten Workflows ein einheitlicher Aufbau mit den drei Zyklen Planung, Durchführung, Iteration erkennbar (siehe Abbildung 3.3). Hierbei muss berücksichtigt werden, dass die Bearbeitung der Workflows keine auf verschiedenen Personen aufgeteilten Rollen kennt. Die Workflows unterscheiden sich hauptsächlich in einem Teilaspekt: Den Einsatz von unterschiedlichen Anwendungen für die Erstellung von Artefakten (z. B. UML-Anwendung für Use-Cases) und der Verwendung teils unterschiedlicher direkt zur Verfügung stehenden Daten. Der Einsatz unterschiedlicher Anwendungen hat aber keine Auswirkung auf den eigentlichen Arbeitsablauf. Weitere Gemeinsamkeiten der Szenarien sind in den folgenden Bereichen zu finden:

- Beratung und Diskussion über Themen
- Präsentationen, die gehalten werden

---

<sup>6</sup>Workflow: Abfolge von Tätigkeiten in einer definierten Reihenfolge, zur Schaffung eines Produkts oder Arbeitsergebnisses. Oft synonym zu Geschäftsprozess verwendet.

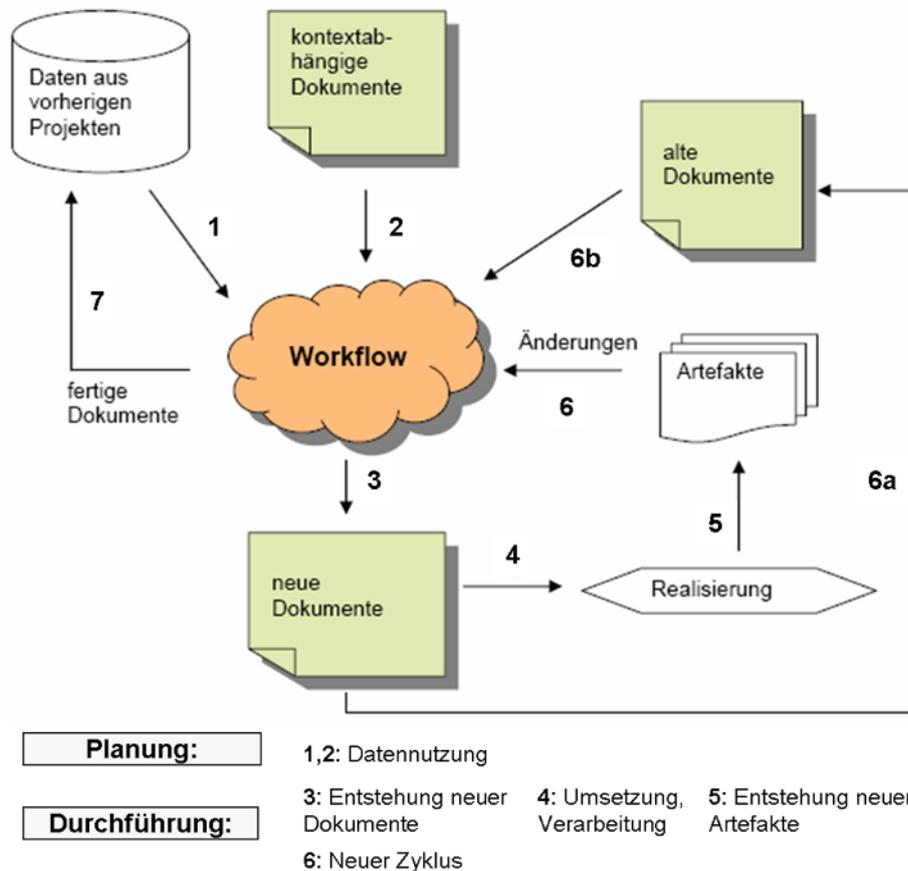


Abbildung 3.3.: Aufbau und Abläufe im Workflow (nach Neuman, 2006, S. 30)

- Brainstorming
- Schreiben von Protokollen
- Zugriffsbeschränkungen
- Erstellung von Zeichnungen und anderen Dokumenten

Eine wichtige aus der Untersuchung resultierende Anforderung an das System war:

*Das CSCW-System muss Werkzeuge bereitstellen, welche die unterschiedlichen Tätigkeiten unterstützt und je nach Workflow die entsprechenden Daten zugreifbar machen.*

Die Gemeinsamkeiten der Szenarien beschränken sich aber nicht nur auf die gleichartigen Workflows und Tätigkeiten, sondern auch auf die darunterliegenden Handlungen und Interaktionen mit dem System. So möchte man z.B. Dokumente für Diskussionsprozesse oder Präsentationen auf der PowerWall anzeigen lassen oder Artefakte in kleiner Gruppe

mit dem Tabletop weiter bearbeiten. Das CSCW-System muss also eine von dem gegenwärtigen Einsatz-Szenario unabhängige Grundfunktionalität bereitstellen. So meint (Bartnik, 2006, S. 18) in seinem Szenario-Entwurf, dass sich allgemeine Funktionen, die in vielen Fachbereichen gebraucht werden, in einem CSCW-System wiederfinden müssen. So sollen u. a.

- Informationen den anderen Teilnehmern einfach und schnell mitzuteilen sein - dies schließt auch die Darstellung der Informationen auf allgemeine sichtbare Flächen mit ein.
- dieselben Dokumente von jedem Teilnehmer eines Projekttreffens zugreifbar sein und gleichzeitig bearbeitet werden können.

Es stellt sich die Frage, ob eine Grundfunktionalität ausreicht, um die weiter oben beschriebenen Workflow adäquat auf Systemebene abzubilden und zu realisieren. Für einen vollständigen Workflow-Ablauf muss das CSCW-System den beteiligten Benutzern die Möglichkeit geben

1. sich beim System anzumelden,
2. Daten für andere bereitzustellen, auszutauschen,
3. Daten gemeinsam zu bearbeiten oder Daten anzeigen zu lassen,
4. Daten zu sichern und
5. sich beim System abzumelden.

Wie könnte eine Realisierung dieser Funktionen unter Beachtung software-ergonomischer Kriterien für CSCW-Systeme aussehen, und sind diese generellen Funktionen auch in den spezielleren Szenarien eines Software-Entwicklungsprozesses effizient anwendbar? Für die einzelnen Funktionen müssen graphische Benutzerschnittstellen entworfen werden, die zu dem definierte, zu einem CSCW-System passende Interaktionsschemata umsetzen.

### 3.2.2. Test-Szenario

Aus der oben angegebenen Liste der Szenarien eines Software-Entwicklungsprozesses wird das Szenario 8 „Zwischenstand“ ausgewählt und näher beschrieben. Die Benutzung dieses Szenario ist natürlich nicht zwingend, da jedes Szenario bedingt durch den repräsentierten **gleichen** Workflow in Frage käme.

### 3.2.2.1. Szenariobeschreibung

Eine Projektgruppe, bestehend aus kleineren Teams, möchte den Projektfortschritt besprechen. Projektteilnehmer die nicht vor Ort sein können, melden sich via Internet an. Bei der Besprechung werden Präsentationen gehalten und die Meilensteindokumente diskutiert. Durch ein Brainstorming<sup>7</sup> soll die Projektarbeit bewertet werden (was läuft gut/schlecht bei der Projektarbeit). Während der Präsentation sammeln die Projektteilnehmer Anmerkungen, welche auf anderen Präsentationsflächen angezeigt werden können. Zum Ende der Besprechung werden, falls erforderlich, Dokumente ausgetauscht.

- Vorbedingung: Erreichen eines bestimmten Projektstatus oder Erreichen eines geplanten (turnusmäßigen) Zeitpunktes für die Besprechung des aktuellen Projektfortschritts.
- Nachbedingung: Verteilung des erstellten Besprechungsprotokoll an alle Teilnehmer sowie Sicherung der Brainstorming-Ergebnisse für die nächste Besprechung.

### 3.2.3. Anwendungsfälle

Aus dem oben beschriebenen Szenario ergeben sich u. a. die folgenden Anwendungsfälle:

#### **Anwendungsfall (A.1): Systemanmeldung**

Der Benutzer meldet sich beim System an oder ab. Die Anmeldung ist im System sichtbar und wird den anderen angemeldeten Benutzern mitgeteilt.

#### **Anwendungsfall (A.2): Bereitstellung von Artefakten**

Der Benutzer organisiert seine für die Sitzung bestimmten Artefakte (Dokumente, Objekte, etc.) in einem speziellen Behälter.

#### **Anwendungsfall (A.3): Verteilung von Artefakten**

Der Benutzer verteilt lokale Artefakte (Dokumente, Objekte, etc.) an andere angemeldete Benutzer oder sendet Artefakte vom Server an angemeldete Benutzer.

#### **Anwendungsfall (A.4): Initiierung eines Bearbeitungsprozess**

Der Benutzer möchte ein lokales Artefakt (z. B. Dokument) mit einem entfernten Gerät und der zum Artefakt-Typ passender Anwendung bearbeiten bzw. betrachten. Hierfür sendet der Benutzer ein ausgewähltes Artefakt zu einem entsprechenden CSCW-Gerät/CSCW-Dienst.

#### **Anwendungsfall (A.5): Administration des Server**

---

<sup>7</sup>Brainstorming: von englisch brainstorm »Geistesblitz«; Gruppendiskussion, bei der spontane Einfälle zu einem bestimmten Problem (z. B. einem Projektvorhaben) gesammelt werden.

Der Administrator konfiguriert das CSCW-System nach den Wünschen der Projektgruppe (z. B. Referenzen zwischen Anwendungstyp und Anwendung) und stellt systeminterne Parameter ein (z. B. Verbindungsparameter für die Dienste-Verwaltung).

#### **Anwendungsfall (A.6): Konfiguration der Clients**

Der Administrator/Benutzer konfiguriert seine Schnittstelle zum System (z. B. lokale Position des Dokumente-Verzeichnisses, Benutzerdaten)

Im Mittelpunkt stehen hierbei die Anwendungsfälle (A.1) - (A.4), die letzten beiden sind der Vollständigkeit halber mit aufgeführt - tatsächlich ist beim späteren Prototyp eine Konfiguration auf beiden Seiten (Client und Server) möglich. Der erste Anwendungsfall A.1 „Systemanmeldung“ wird dem Prinzip der „Awareness“ gerecht - die Systemabmeldung wurde hierbei vernachlässigt (ähnelt aber der Anmeldung, bezogen auf die Bekanntmachung neu angemeldeter Benutzer). Detailliertere Beschreibungen der einzelnen Anwendungsfälle stehen im Anhang A.

### **3.3. Die Idee**

Dem Benutzer soll es ermöglicht werden, von jedem Platze respektive Gerät aus Einfluss auf die aktuelle Diskussion zu nehmen. In Bezug auf das oben genannte Szenario, kann sich ein Projektteilnehmer mit seinem eigenen Laptop oder ein im CSCW-Raum zur Verfügung stehendem Gerät anmelden und, wenn gewünscht, einen bestimmten Einfluss, beispielsweise über eine entfernte Steuerung von anderen CSCW-Geräten, auf den aktuellen Diskussionsprozess ausüben. Der Projektteilnehmer erhält somit für eine bestimmte Zeitspanne die Kontrolle über eines der im Raum zur Verfügung stehenden CSCW-Geräte. Dieses beschreibt im Wesentlichen die Idee. Ein Projektteilnehmer kann sich mit einem oder mehreren CSCW-Geräten verbinden und diese für Präsentationszwecke (z. B. Anzeigen von Bildern, Graphen, etc.) nutzen oder Dokumente für die weitere gemeinsame Bearbeitung zur Verfügung stellen. Die Abbildung: 3.4 „Interaktiver Gruppenarbeitsraum - Skizze“ soll dieses verdeutlichen. Die in der Abbildung gezeigten Markierungen (1 - 4) stehen für mögliche Aktivitäten der angemeldeten Benutzer im Raum:

1. Stellt den Vortragende dar (beispielsweise bei einer Präsentation), der aktiv das große interaktive Display benutzt, gleichzeitig aber mit einem kleineren Display verbunden ist, um hier zusätzliche Informationen darzustellen.
2. Hier möchte ein Benutzer zur aktuellen Diskussion Informationen liefern und diese den anderen Teilnehmern auf einem kleineren Display präsentieren.

3. Wie 2., nur ist dieser Teilnehmer als einziger mit diesem Display (respektive CSCW-Gerät) verbunden.
4. Dieser Teilnehmer möchte parallel zum Vortragenden Informationen auf dem großen interaktiven Display darstellen oder bearbeiten.

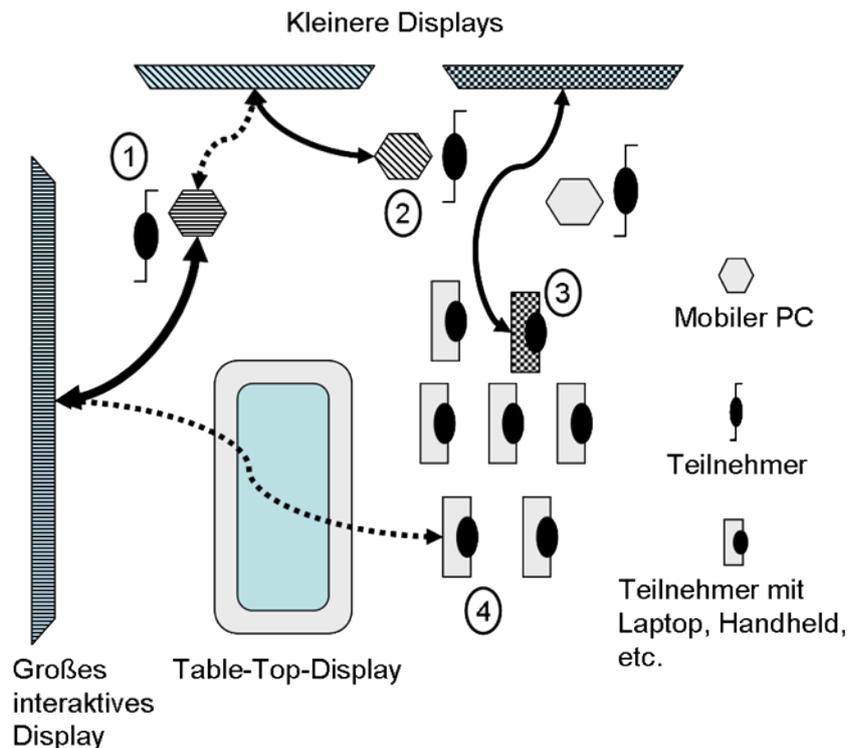


Abbildung 3.4.: Interaktiver Gruppenarbeitsraum - Skizze

Wie könnte eine mögliche Kontrolle anderer Geräte aussehen, welche Interaktionsformen gibt es oder sind denkbar? Das nachfolgende Kapitel beschreibt zwei Interaktionsarten (oder auch Ideen), die eine entfernte Kontrolle/Steuerung von CSCW-Geräten respektive -Software realisieren.

### 3.3.1. Kontrolle und Steuerung entfernter Geräte

Die beiden Arten respektive Systeme, welche die Steuerung/Kontrolle von entfernten Geräten realisieren, unterscheiden sich prinzipiell in der Sicht des virtuellen Arbeitsplatzes oder auch Desktops.

### 3.3.1.1. Virtual Network Computing:

Eine Virtual-Network-Computing-Software (VNC) zeigt den Bildschirminhalt eines entfernten Rechners (der VNC-Server) auf einem lokalen Rechner (der VNC-Client) und überträgt lokale Tastatur- und Mausereignisse (VNC-Client) zum entfernten Rechner (VNC-Server), welche sogleich zur eigenen Event-Queue<sup>8</sup> weitergeleitet werden (siehe Abbildung: 3.5). Der Benutzer kann somit auf einem entfernten Rechner arbeiten, als säße er davor. Hierbei werden die Desktops oder Arbeitsflächen der beiden Rechner streng unterschieden, mit der Konsequenz das Desktop übergreifende Interaktionen, wie zum Beispiel das nahtlose Verschieben von Dokumenten vom lokalen Desktop auf den entfernten Desktop, nicht möglich ist<sup>9</sup>. Der entfernte Rechner wird als eigenständiger Rechner „gesehen“ und hat nichts mit der Anwendungsdomäne des lokalen Rechners (wie zum Beispiel Desktop) zu tun.

VNC-Software benutzen das RFB-Protokoll (Remote Frame Buffer) und sind plattformunabhängig (d. h. VNC-Server und -Client können auf unterschiedlichen Plattformen laufen; klare Trennung zwischen Kommunikationsschicht und den anderen Schichten), unterscheiden sich aber in ihrem Funktionsumfang, z. B. Verschlüsselung, Audio-Unterstützung und Mehrfach-sitzungen. Die Bezeichnungen VNC und RDS (Remote Desktop Software) werden oftmals synonym verwendet; RDS kann auch als Überbegriff angesehen werden, da RD-Software unterschiedliche Protokolle u. a. auch VNC einsetzt.

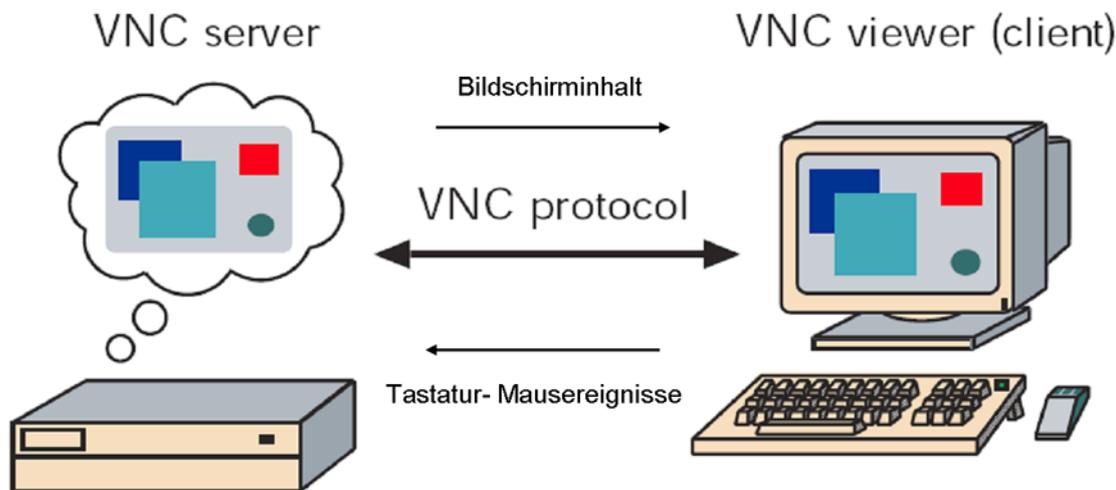


Abbildung 3.5.: Virtual Network Computing

<sup>8</sup>Event-Queue: engl. event = Ereignis, engl. queue = Schlange, Warteschlange für Ereignisse

<sup>9</sup>tatsächlich unterstützen mittlerweile einige VNC-Programme den Datei-Transfer in irgendeiner Art und Weise  
- dennoch besteht eine klare Trennung zwischen den beteiligten Rechner

Für das anfangs genannte Szenario hat der Einsatz von VNC-Software Vor- und Nachteile:  
Vorteile:

- **Unabhängigkeit der Ausrichtung:** Die Ausrichtung des lokalen Desktops (Nord, Süd, Ost, West) ist von der Position des entfernten Desktops unabhängig. Der Laptop kann seitlich oder in entgegengesetzter Richtung zum entfernten Desktop ausgerichtet sein, ohne dass die An-/Steuerung des entfernten Desktop verwirrend wäre (vgl. „Gemeinsamer Arbeitsplatz/Desktop“).
- **Anzahl der Desktops:** Die Anzahl parallel laufender entfernter Desktops ist theoretisch nicht begrenzt. Praktisch muss man die Anzahl auf ein bestimmtes Maß beschränken, da die Handhabbarkeit des eigenen Desktops (überladen mit kleineren Fremd-Desktops) bei einer wachsenden Anzahl entfernter Desktops sinkt.

Nachteile:

- **Inkonsistentes Raum-Modell:** Die Abgrenzung des eigenen Arbeitsbereiches (Desktop) von den Arbeitsbereichen der entfernten Desktops (Powerwall, Seitendisplays, etc.) stört das mentale Modell eines „gemeinsamen Desktops“.
- **Hohe Netzwerklast:** Die Netzwerklast erhöht sich mit jedem neu angemeldeten Benutzer und den verbundenen entfernten Desktops (Datenübertragung der Desktop-Bilder).

### 3.3.1.2. Gemeinsamer Arbeitsplatz/Desktop:

Diese Möglichkeit der Steuerung entfernter Geräte ist stark im ubiquitären Bereich angesiedelt. Die Idee dahinter ist, dass angemeldete Geräte ihren Arbeitsplatz/Desktop global zur Verfügung stellen können - der Desktop quasi vergrößert wird - oder ihren eigenen Desktop erweitern. Bewegt sich z. B. der Mauszeiger über den lokalen Bildschirmrand hinaus, so taucht dieser auf einem anderen Desktop wieder auf - und der Benutzer kann wie gewohnt dort weiterarbeiten. Die Abbildung: 3.6 zeigt die unterschiedlichen Desktops der untereinander verbundenen Anwender und eine mögliche Interaktion. Unter (1) und (2) sind die Desktopübergänge von Desktop 1 (Südkante) und 2 (Nordkante) respektive Desktop 2 (Ostkante) und 3 (Westkante) zu sehen. (3) ist ein Spezialfall, da hier nicht klar ist, wo ein Übergang vom Desktop 3 zu Desktop 1 zu erfolgen hat - spätestens hier ist das mentale räumliche Bild vom gemeinsamen Desktops gestört. Das Problem ließe sich dadurch umgehen, dass nur eine gerade Anzahl von Desktops zugelassen wird und die geformte Arbeitsfläche rechteckig ist. (4a) und (4b) zeigen die Bewegung eines Objektes (von Desktop 2) über den Desktoprand hinaus. Der Anwender versucht seine Vorstellung des gemeinsamen Desktop mit der realen Umgebung abzugleichen. Hierbei müssen im Wesentlichen drei Punkte beachtet werden:

1. Die Anzahl der Anzeigegeräte respektive Desktops: Eine hohe Anzahl könnte für den Anwender hinsichtlich der Interaktion verwirrend sein. Wo befindet sich beispielsweise der Übergang zu dem fünften Desktop, wenn mehr als vier Desktops miteinander verbunden sind und nur eine Transition von dem lokalen Desktop auf ein „Fremddesktop“ erlaubt ist (ein Desktop kann nur durch vier andere erweitert werden)?
2. Die räumliche Anordnung der Anzeigegeräte: Sind die Anzeigegeräte in einer rechteckigen Form angeordnet, so deckt sich dieses mit dem mentalen Modell des Anwenders von einem gemeinsamen Desktop. Ist der gemeinsame Desktop nur „lose“ aufgebaut (z.B. drei Laptops stellen gegenseitig ihre Desktops zur Verfügung, sind aber im Raum verteilt), könnte dieses für den Benutzer verwirrend sein.
3. Ziel des gemeinsamen Desktops: Bereitstellung einer größeren Fläche für die gemeinsame Bearbeitung von Aufgaben oder Bereitstellung einer einfachen Navigation<sup>10</sup> zu „Fremd-Desktops“ und deren Steuerung?

Ein gemeinsamer Desktop, bestehend aus mehreren rechteckig angeordneten Displays und der dahinter liegenden Steuerungshardware, ist u. a. mit der Powerwall realisiert - hier aber mit einer statischen Anzahl von Displays, d.h. das dynamisches Erweitern des Desktops durch andere Benutzer ist nicht vorgesehen; die Powerwall steht für sich alleine. Vor- und Nachteile für den Einsatz eines „gemeinsamen Desktops“ in dem weiter oben genannten Szenario wären:

Vorteile:

- Mentales räumliches Modell: Das Modell eines gemeinsamen Desktop respektive einer Arbeitsfläche wird gewahrt. Der Benutzer kann den Mauszeiger o. ä. intuitiv durch den gemeinsamen Desktop bewegen.
- Verschieben von Anwendungen: Ist eine Verbindung mit einem entfernten Desktop aufgebaut, so kann der Benutzer ein lokales Anwendungsfenster oder Dokument von seinem Desktop über die Verbindungskante zum entfernten Desktop verschieben - andere angemeldete Benutzer können dann diese Anwendung respektive dieses Dokument benutzen.

Nachteile:

- Begrenzte Anzahl der Desktops: Bei der Benutzung von Geräten mit rechteckiger Anzeigefläche (Laptop, Handheld) ist die Anzahl der entfernten Desktops auf vier begrenzt, da nur vier Seiten für eine Verbindung/Erweiterung zur Verfügung stehen.

---

<sup>10</sup>Mit der Open-Source Software „Synergy“ lassen sich die Desktops unterschiedlicher Ein- und Ausgabegeräte über die Desktopränder miteinander verbinden. Hier werden nur Tastatur- und Mausereignisse des lokalen Systems beim Übertritt in das andere System entsprechend weitergeleitet - Objekte, Dokumente, etc. können nicht über Desktopgrenzen hinweg verschoben werden.

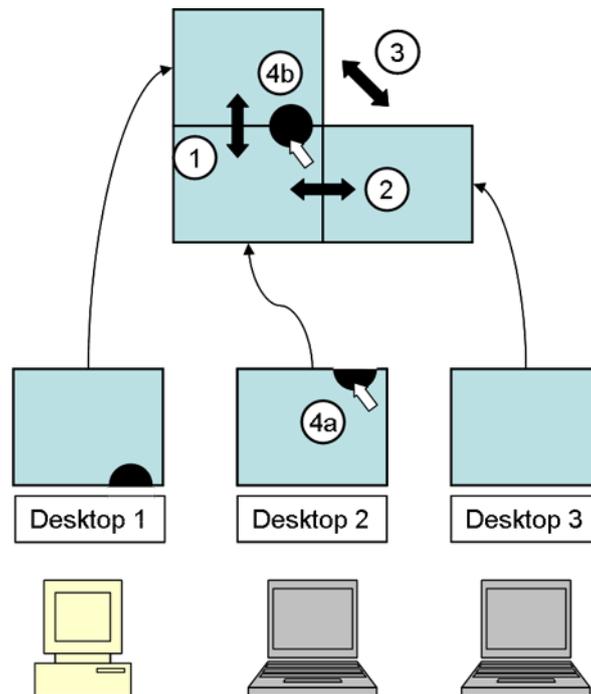


Abbildung 3.6.: Gemeinsamer Desktop/Arbeitsplatz - Darstellung und Interaktion: (1)-(3) mögliche Übergänge, (4a) u. (4b) Bewegung eines Objekts v. Desktop 2 zu Desktop 1

- **Abhängigkeit von der Ausrichtung:** Bei der Erweiterung oder Verbindung lokaler Desktops durch bzw. mit entfernten Desktops muss immer die Ausrichtung der Anzeige-Geräte berücksichtigt werden. Befindet sich beispielsweise rechts vom Benutzer ein entfernter Desktop (in Form eines Touchscreen), so wird er diesen immer mit der rechten Desktop-Kante verbinden. Dreht sich der Benutzer mit seinem Eingabegerät um 90 Grad nach rechts, stimmt der Kantenübergang zum entfernten Desktop nicht mehr mit der Ausrichtung dieses Gerätes überein. Intuitiv würde der Benutzer versuchen, den Mauszeiger über die nördliche Kante zu bewegen, um auf den entfernten Desktop zu gelangen, müsste aber die östliche Kante wählen, da dort der voreingestellte Übergang ist.

Eine interessante Variante in Bezug auf die Navigation durch einen gemeinsamen Desktop, stellt die in Abbildung: 3.7 dar. Der Benutzer des Laptops kann mit Hilfe eines virtuellen Fensters durch den gemeinsamen Desktop „navigieren“.

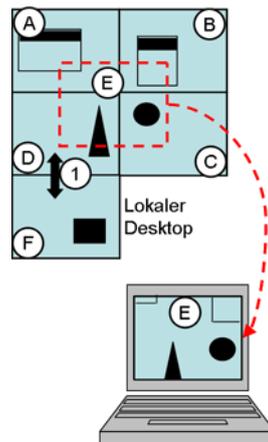


Abbildung 3.7.: Gemeinsamer Desktop/Arbeitsplatz - Navigation: (A)-(D) stellt eine Powerwall dar, (F) ist der Original Desktops des Anwenders, (E) Fensterausschnitt des gemeinsamen Desktop, (1) möglicher Übergang Original-Desktop/Powerwall-Desktop

Ein Beispiel für eine spontane Erweiterung von Desktops zeigt das System der „Bumping Displays“ von Ken Hinckley. Zwei oder mehrere Tablet-PCs<sup>11</sup> (kurz Tablets) können beliebig über ihre vier Kanten verbunden werden (sie stoßen dabei zusammen; engl. to bump). Das System erkennt zusätzlich die Ausrichtung der beteiligten Tablets: Werden zwei Tablets gegenüberliegend zusammengebracht (die beiden oberen Kanten der Tablets), so wird in den Face-To-Face Kollaborationsmodus gewechselt. Die Abbildungen: 3.8 und 3.9 zeigen die Anwendungsarten der Tablets.

### 3.3.2. Multi-Desktop-Steuerung

#### 3.3.2.1. Multi-Desktop-Steuerung: Gesamtbild

Die Multi-Desktop-Steuerung (MDS) soll die Vorteile beider Steuerungsarten, VNC und „gemeinsamer Desktop“, in sich vereinen. Die Abbildung 3.10 (S.71) zeigt den CSCW-Raum mit diversen Ein- und Ausgabegeräten aus der Sicht des Benutzers. Wie bei VNC sollen unterschiedliche Displays und die dort angezeigten Desktops von dem Benutzer gesteuert werden können. Gleichzeitig sollen die entfernten Desktops in dem lokalen Desktop des Be-

<sup>11</sup>Tablet-PC: ein mobiler Computer, der das Format eines Schreibblocks aufweist und wie eine Schreibtafel bedient wird. Seine Oberseite wird von einem berührungsempfindlichen Bildschirm gebildet, über den man die Daten per Stift eingeben kann.

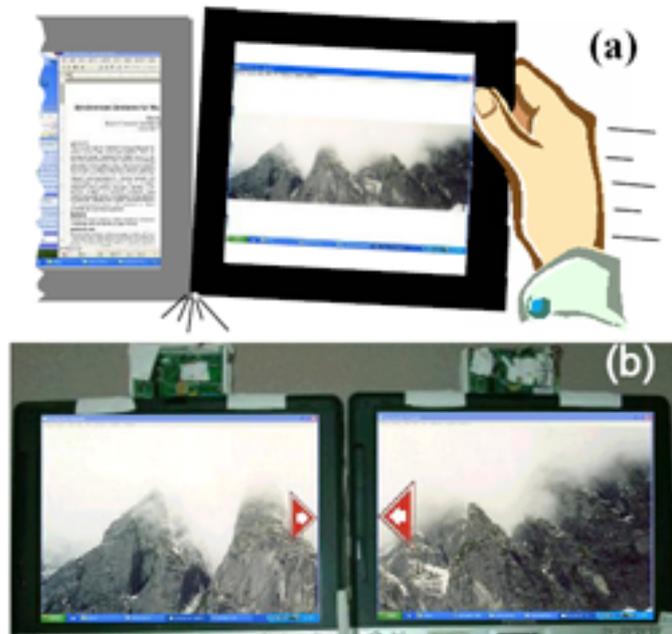


Abbildung 3.8.: Bumping Displays 1 - (a) Kacheln dynamischer Displays durch das Zusammenstoßen (bumping) zweier Tablets. (b) Die Tablets bilden zeitweilig ein größeres Display und ermöglichen dadurch die Anzeige eines größeren Fotos (Gebirge). Die kleinen roten Pfeile signalisieren dem Benutzer (Feedback) die Verbindung beider Tablets bei diesen Kanten.

nutzers „integriert“ sein. (a)<sup>12</sup> Die Inhalte der im Gruppenarbeitsraum installierten Displays (Powerwall, Display 1 Display 2) werden VNC-ähnlich auf dem Desktop des angemeldeten Benutzers verkleinert angezeigt und fortwährend aktualisiert. Auf den Desktops befinden sich bereits geöffnete Anwendungen (b) oder Dokumente (c). Bei den nachfolgenden dargestellten Interaktionen wird davon ausgegangen, dass sich die Benutzer via Laptop oder Personal Computer anmelden - Geräte die eine ausreichend große Anzeigefläche bieten. Kleinerer Anzeige-Geräte wie Handheld, Handy<sup>13</sup> werden hier nicht weiter betrachtet.

### 3.3.2.2. MDS - Dokumente verschieben

Der Benutzer soll die Möglichkeit haben, lokale auf seinem Desktop befindliche Dokumente auf anderen im CSCW-Raum zur Verfügung stehenden Ausgabegeräte zu verschieben. Im Vordergrund steht die einfache intuitive Interaktion mit dem entfernten Gerät. Abbildung

<sup>12</sup>Die von Paranthesen umgebenden Buchstaben oder Zahlen beziehen sich immer auf die zuvor erwähnte Abbildung

<sup>13</sup>eher englisch Mobile zu nennen

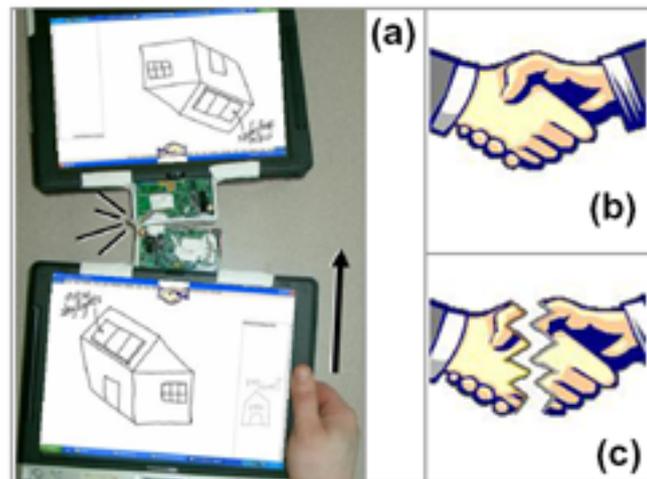


Abbildung 3.9.: Bumping Displays 2 - (a) Hier sieht man eine Face-To-Face (von Angesicht zu Angesicht) Kollaboration durch das Zusammenführen beider Tablets. Die abgebildete Skizze (Haus) mit Notizen der Benutzer wird korrekt ausgerichtet. (b),(c) Die Benutzer bekommen eine klare Rückmeldung in Form von aussagekräftigen Piktogrammen über die Verbindung oder spätere Unterbrechung der kollaborativen Sitzung.

3.11 (S.72) zeigt unter (1a) den Beginn der Interaktion (der Desktop des Laptops ist nur ausschnittsweise zu sehen), (1b) den Status beider Desktops nach der Interaktion. Der Desktop von „Display X“ ist verkleinert auf den Desktop des Benutzers zu sehen (c). Der Benutzer verschiebt ein Dokument mit der bekannten „Drag-And-Drop<sup>14</sup>“-Interaktion in den verkleinert dargestellten Ziel-Desktop (1a)(a). (1b)(a) zeigt die beiden Desktops nach der Aktion: Das Dokument ist auf dem Desktop von „Display X“ zu sehen. Wichtig hierbei ist, dass die Position des Dokumentes im Display X ungefähr der Position entspricht, welche der Mauszeiger vor dem Drop-Vorgang über den verkleinert Desktop hatte (korrekte Positionierung der Dokumente auf den Ziel-Desktops).

### 3.3.2.3. MDS - Verschieben geöffneter Anwendungen

Geöffnete Anwendungen und die darin in Bearbeitung befindlichen Dokumente, sollen nahtlos auf andere öffentliche Ausgabegeräte verschoben werden können - dargestellt in Abbil-

<sup>14</sup>Drag-And-Drop = Ziehen und Ablegen: Ein Verfahren zum Verschieben und Kopieren beliebiger Informationen, etwa eines Texts, eines Bilds, eines Icons oder eines Verzeichnisses. Das betreffende Objekt muss zunächst markiert werden (etwa mit den Cursor-Tasten oder per Mausklick) und wird danach bei gedrückter Maustaste an die neue Position gezogen und hier durch Loslassen der Taste abgelegt. Das Verfahren hat sich mittlerweile etabliert und wird deshalb schon **intuitiv** eingesetzt.

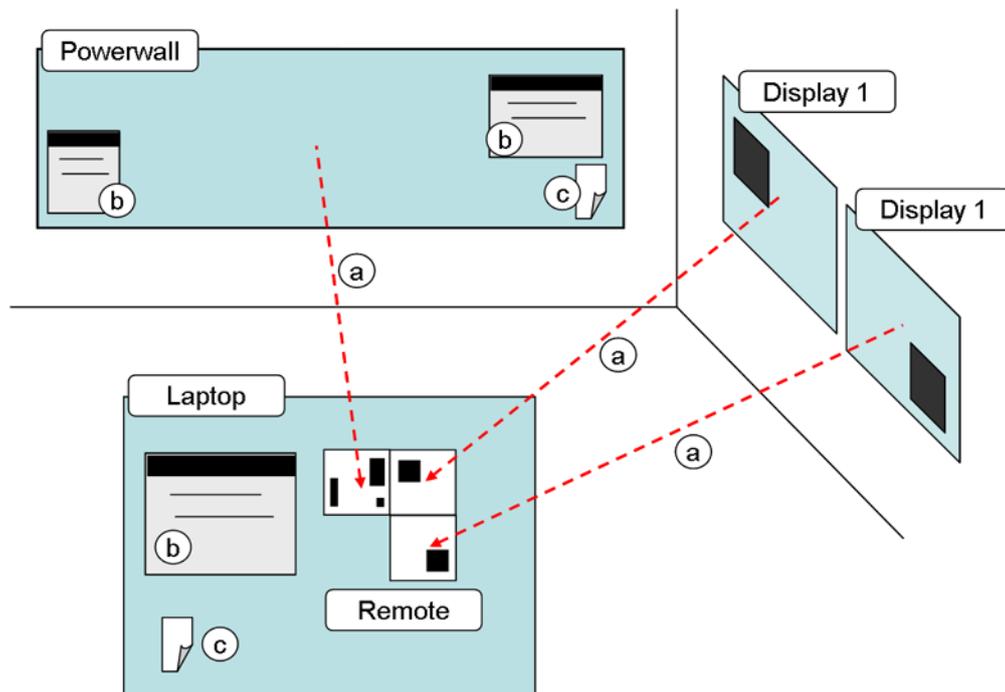


Abbildung 3.10.: MDS - Gesamtbild: (a) Entfernte Desktop, (b) Geöffnete Anwendungen, (c) Dokumente

Abbildung 3.12 (S.73). Im Wesentlichen ähnelt sie der zuvor beschriebenen Interaktion, nur dass dieses Mal Anwendungsfenster verschoben werden (2a)(a). Auch hier entspricht die spätere Position des Anwendungsfenster bei dem entfernten Desktop, der zuvor anvisiert in der verkleinerten Darstellung. Die Größe des Anwendungsfenster wird beibehalten (2b)(a).

#### 3.3.2.4. MDS - Dokumente verschieben und öffnen

Das Verschieben von Dokumenten ist bereits weiter oben beschrieben worden. Hier wird zusätzlich eine adäquate Anwendung auf dem Ziel-Desktop gestartet und das dorthin verschobene Dokument geöffnet. Abbildung 3.13 (S.74) zeigt unter (3a)(a) die Desktops vor und unter (3b)(a) (mit gestarteter Anwendung) nach dem Drop-Vorgang. Die in der Abbildung gezeigte Interaktion bezieht sich auf den Anwendungsfall A.4 (S.110) „Initiierung eines Bearbeitungsprozesses“.

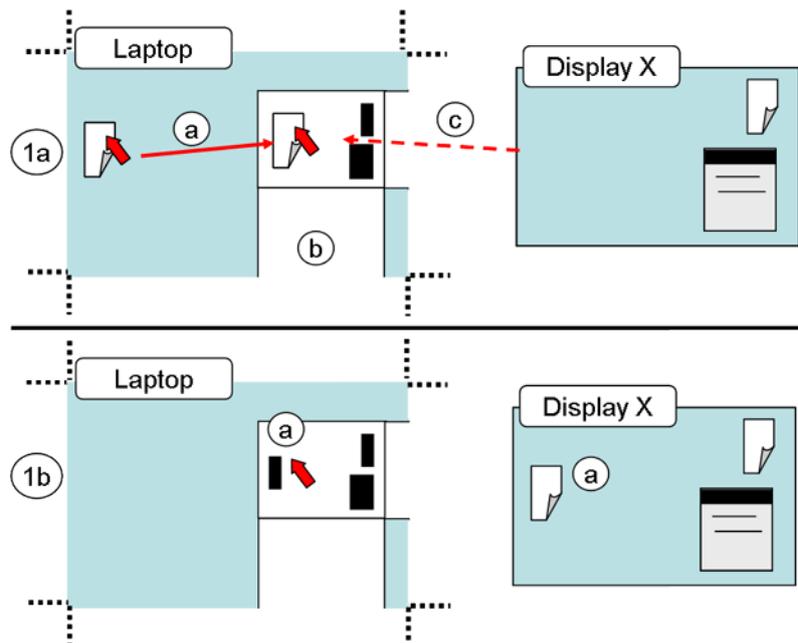


Abbildung 3.11.: MDS - Verschieben 1: Verschieben eines Dokumentes vom lokalen Desktop (Laptop) zu einem entfernten Desktop (Display X).

### 3.3.2.5. MDS - Anmeldung

Die Benutzer sollen sich bei den unterschiedlichen im CSCW-Raum befindlichen Ausgabegeräten anmelden können. Die Anmeldung soll hierbei für jeden angemeldeten Benutzer sichtbar sein. Die Abbildungen 3.14 (S.74) und 3.15 (S.75) verdeutlichen dieses. Zwei Teilnehmer möchten sich gerne beim „Display X“ anmelden (1a)(a). In (1b)(c) sind die jeweiligen Benutzer auf den jeweils anderen Desktop abgebildet. „Display X“ hat dementsprechend Abbildungen o. ä. beider Benutzer auf seinem Desktop (a)(b). Die Abbildungen sollen später als Interaktionskomponenten für die Verteilung von Dokumenten dienen.

### 3.3.2.6. MDS - Verteilung von Dokumenten

Jeder angemeldete Teilnehmer ist mit einer Abbildung o. ä. auf den Desktop der anderen Teilnehmer und der entfernten (steuerbaren) Desktops zu sehen (Abbildung 3.16 (S.75)). Auf dem Desktop des „Teilnehmer 1“ ist

- die Abbildung von „Teilnehmer 2,“ und
- eine Verkleinerung des entfernten Desktops von „Display X“ (b)

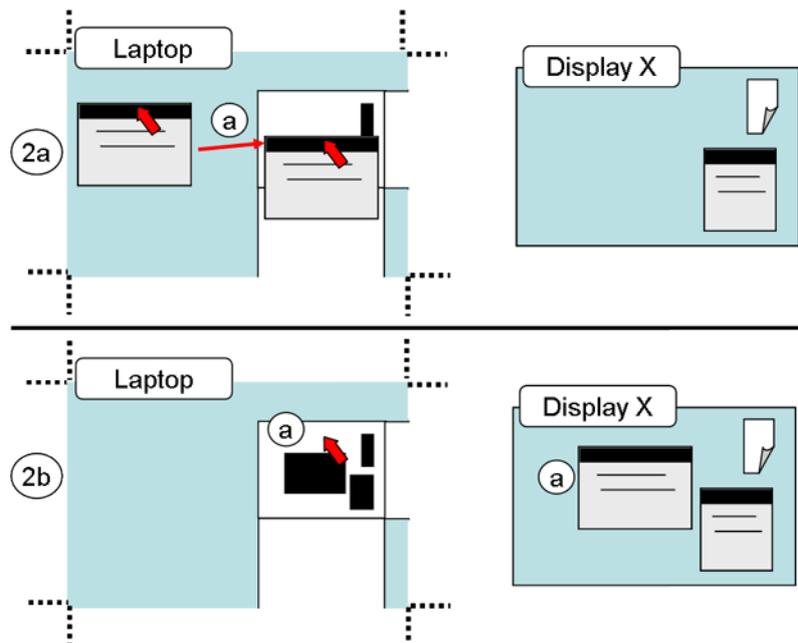


Abbildung 3.12.: MDS - Verschieben 2: Verschieben einer geöffneten Anwendung

dargestellt - entsprechendes ist auch auf den Desktop von „Teilnehmer 2“ zu sehen. „Teilnehmer 1“ möchte „Teilnehmer 2“ ein Dokument zusenden. Diese geschieht über eine Drag-And-Drop-Aktion (a):

- „Teilnehmer 1“ Wählt das Dokument und
- zieht dieses auf die Abbildung o. ä. von „Teilnehmer 2“ .

Das Dokument taucht auf den Desktop (die Position wird anders als bei den vorigen Verschiebe-Aktionen vom System bestimmt) des zweiten Teilnehmers auf (c).

### 3.3.3. MDS - Grenzen der Realisierung

Wie im Kapitel 3.1 „Rahmen und Abgrenzung“ bereits erwähnt, kann der Entwurf der „Multi-Desktop-Steuerung“ nicht bis in das kleinste Detail realisiert werden - dieses würde den Rahmen dieser Arbeit sprengen. Die tatsächlich Fernsteuerung, bekannt aus VNC-Software oder „Remote-Desktop“-Anwendungen, wird zum großen Teil nur simuliert. Die zuvor aufgezeigten Interaktionen, unterlegt durch die Anwendungsfälle (siehe Anhang A), finden sich im Prototypen wieder.

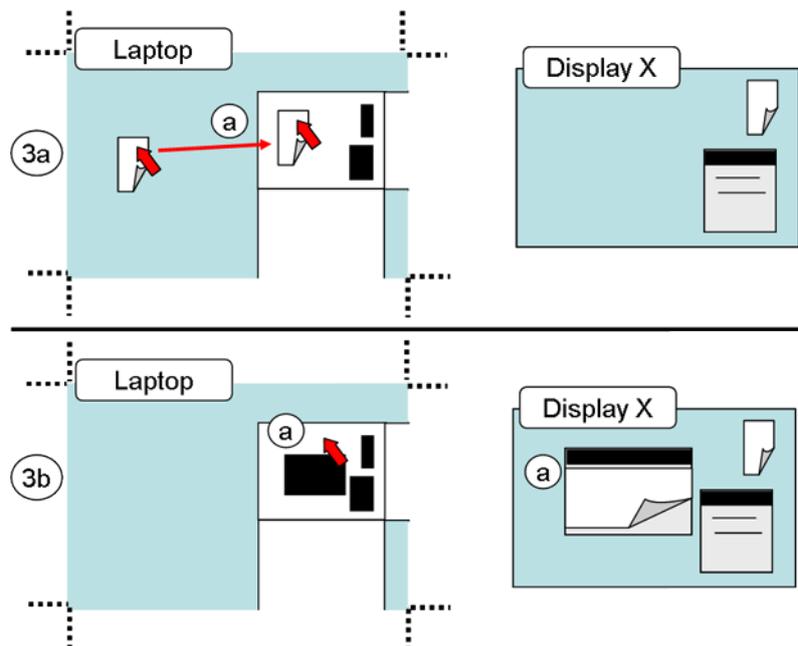


Abbildung 3.13.: MDS - Verschieben 3: Verschieben eines Dokumentes und Öffnen mit einer adäquaten Anwendung bei dem Ziel-Desktop.

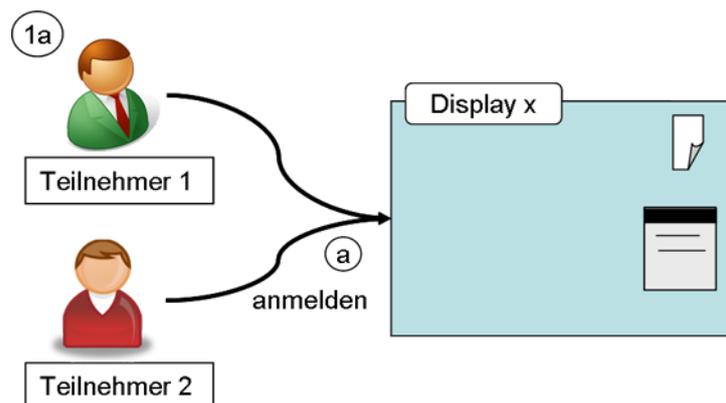


Abbildung 3.14.: MDS - Anmeldung 1: Zwei Benutzer möchten sich bei einem entfernten Desktop anmelden.

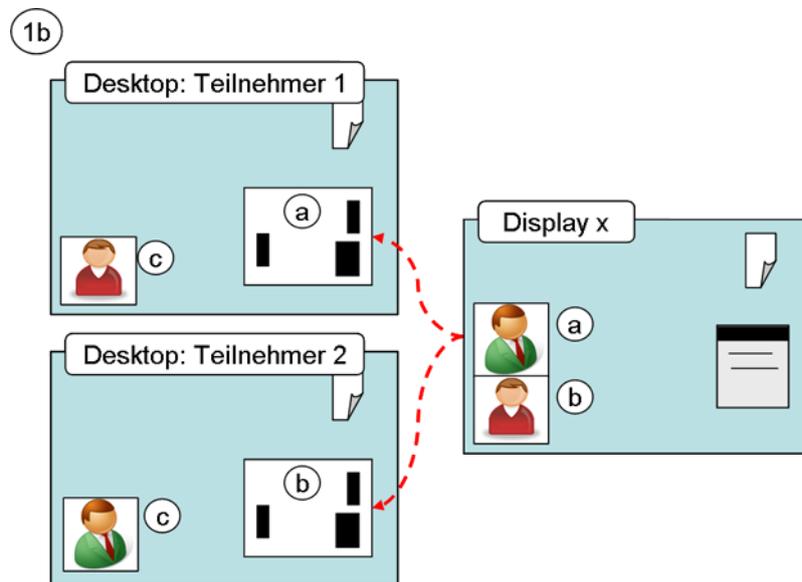


Abbildung 3.15.: MDS - Anmeldung 2: Die Benutzer sind angemeldet.

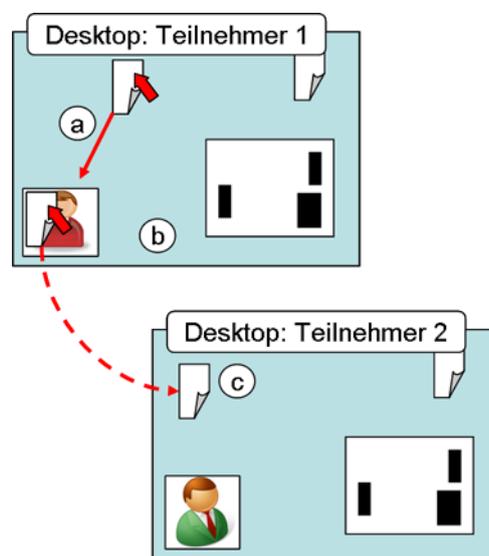


Abbildung 3.16.: MDS - Verteilung: Teilnehmer 1 sendet ein Dokument an Teilnehmer 2.

## 4. Prototyp - „ProjectRoom“

Mit dem Prototyp soll in erster Linie die Kernfunktionalität und die einhergehenden Interaktionen getestet werden, welche ein CSCW-System zur Verfügung stellen muss. Die Architektur und das spätere Design ist einfach gehalten und dient lediglich der schnellen Umsetzung der analysierten Anforderungen. Der Schwerpunkt liegt auf die grafische Benutzeroberfläche der Benutzer-Anwendung (Client). Folgende Bereiche sind bei dem Entwurf des Prototyps vernachlässigt worden oder kommen nur in einem begrenzten Maße vor:

1. **Projekterwaltung:** Für eine gemeinsame Bearbeitung von Aufgaben, ist eine Verwaltung der Projekte unerlässlich. Eine derartige Verwaltung gibt es hier nicht.
2. **Persistenz:** Wie und wo sollen die Daten gesichert werden? Die Sicherung der Daten erfolgt beim Prototyp lokal, sowohl bei der Client-Anwendung als auch beim Server. Die Daten liegen somit als unabhängige verteilte Kopien vor - mit der Gefahr der Dateninkonsistenz. Die Daten werden nur über das Datei-System verwaltet, die Nutzung eines DBMS<sup>1</sup> ist nicht vorgesehen.
3. **Rechteverwaltung:** Eine Rechteverwaltung, welche den unterschiedlichen Benutzern Rollen zuteilt und den Rollen gemäß Rechte auf Objekte etc. vergibt, gehört nicht zum Funktionsumfang des Prototyps. Tatsächlich gibt es nur zwei Rollen: Der Benutzer der Client-Anwendung und der Administrator für den Serverdienst.
4. **Sicherheit:** Die Sicherheit in Form einer Authentifizierung des Benutzers im System wird simuliert. Tatsächlich dient die Anmeldung der Benutzer vor allem der Bekanntmachung ebendieser im System und natürlich den Verbindungsaufbau zu den einzelnen Servern.
5. **Dienste-Verwaltung:** Der Entwurf der Dienste-Verwaltung ist einfach gestaltet.

In einer Gesamtarchitektur eines CSCW-System würde jeder der einzelnen Punkte durch einen eigenen Server<sup>2</sup> (und dessen innere Architektur) abgebildet werden. Eine umfassende Architektur und Entwurf eines CSCW-System ist in der Diplomarbeit von Roman Bartnik „Weiterentwicklung einer Technologiebasis für interaktive Gruppenarbeitsräume“ zu finden.

---

<sup>1</sup>Datenbankmanagementsystem

<sup>2</sup>Projektservers, Datenbankserver, Sicherheits-Server, Dienste-Verwaltungs-Server

Die nachfolgenden Kapitel beschreiben den Entwurf und die Realisierung des Prototyps. Danach erfolgen die Beschreibungen der einzelnen Interaktionen.

## 4.1. Entwurf und Realisierung

### 4.1.1. Architektur des Systems

Die Architektur des Prototyps ist eine typische Client/Server-Architektur für verteilte Anwendungen. Hauptanforderung war es, eine dynamische Bindung von Clients zu Servern oder Services<sup>3</sup> zu gewährleisten. Die Clients sollen jederzeit diverse Dienste in Anspruch nehmen können. Hierfür wird eine Dienste-Verwaltung benötigt, welche die Dienste eines Servers veröffentlicht, verwaltet und für die Clients eine Funktionalität bereitstellt, welche Service-Findung, Service-Verbindung und Service-Trennung beinhaltet. In Abbildung 4.1 (S.78) ist die Service-Verwaltung abstrakt als Service-Bus dargestellt. Unterschiedliche Server S1-S3 können ihre Dienste auf den Service-Bus veröffentlichen (1). Diese können wiederum von den Clients auf dem Bus lokalisiert (2) und abonniert werden (3). Die Clients können mehrere Dienste simultan in Anspruch nehmen.

Die Kommunikation der einzelnen Komponenten untereinander ist in Abbildung 4.2 (S.79) dargestellt und konkretisiert die Architektur des Systems. Die eingesetzte Hardware deckt sich zum Teil mit der Gesamtarchitektur des Systems, d.h. Server müssen, bedingt durch die Ansteuerung der Displays, auf einer Server-Hardware laufen.

- **Server:** (1) Der Server kontrolliert Anwendungen und mindestens ein (2) öffentliches Display (und den Desktop auf der logischen Seite). (3) Er ist für die Verteilung von Desktops, Dokumente und Anmeldeinformationen an die Clients zuständig.
- **Client:** (4) Die Clients bekommen einzelne Desktops, Anmelde-Informationen etc. von den Servern und senden ihrerseits Dokumente respektive Steuerkommandos an die Server (5).

Insgesamt besteht der Prototyp aus fünf Komponenten:

1. **Server:** Room-Server - Stellt Services und eine graphische Benutzerschnittstelle (GUI<sup>4</sup>) für die Bedienung des angebotenen öffentlicher Displays bereit.
2. **Service:** Room-Service - Stellt Funktionalität für den Client bereit.
3. **Kommunikation:** Sorgt für die Kommunikation der Komponenten untereinander.

---

<sup>3</sup>Service = Dienst, konzeptionelle Sicht

<sup>4</sup>Graphical User Interface = grafische Benutzerschnittstelle

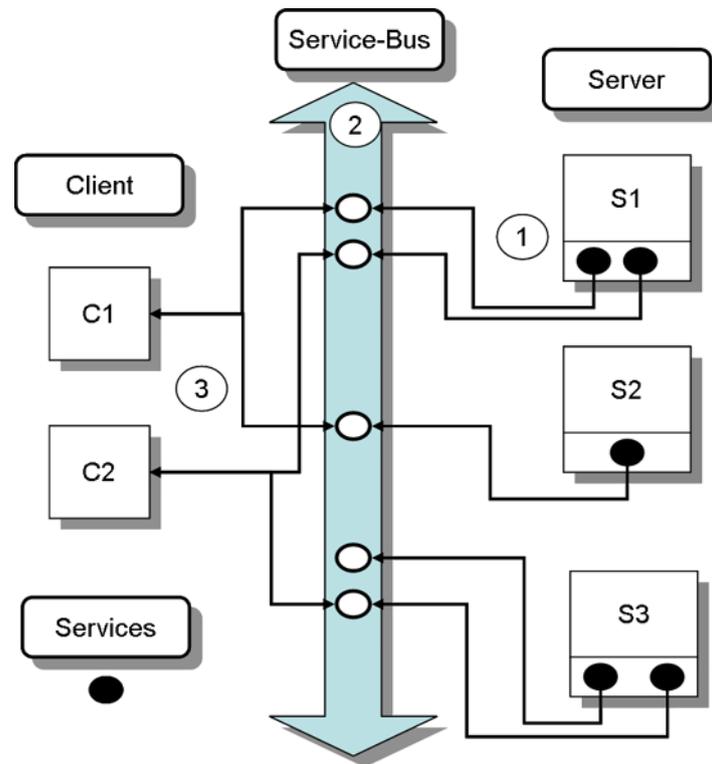


Abbildung 4.1.: Entwurf - Client/Server-Architektur mit Service-Bus/Verwaltung

4. **Client:** Room-Client - Hauptanwendung für den Projektteilnehmer, stellt über eine GUI Steuerungskomponenten im System bereit.
5. **Service-Verwaltung:** Verwaltet die von Room-Servern angebotene Dienste: Service-Veröffentlichung, Service-Findung, Service-Trennung

Bei Room-Server, Room-Client und Room-Service finden hauptsächlich zwei Architekturmuster Anwendung:

- **MVC-Muster<sup>5</sup>:** Sorgt für klare Verantwortlichkeiten der einzelnen Komponenten.
- **Beobachter-Entwurfsmuster:** Sorgt für eine schwache Koppelung der einzelnen Komponenten im MVC-Modell, mit dem Vorteil, dass diese leicht ausgetauscht werden können.

Abbildung 4.3 (S.80) zeigt die einzelnen Komponenten in ihrer MVC-Architektur. (1) und (2) stellt die Kommunikation der beiden anderen mit der Service-Komponente dar. (3) Alle drei

<sup>5</sup>Model View Controller = Modell, Präsentation, Steuerung

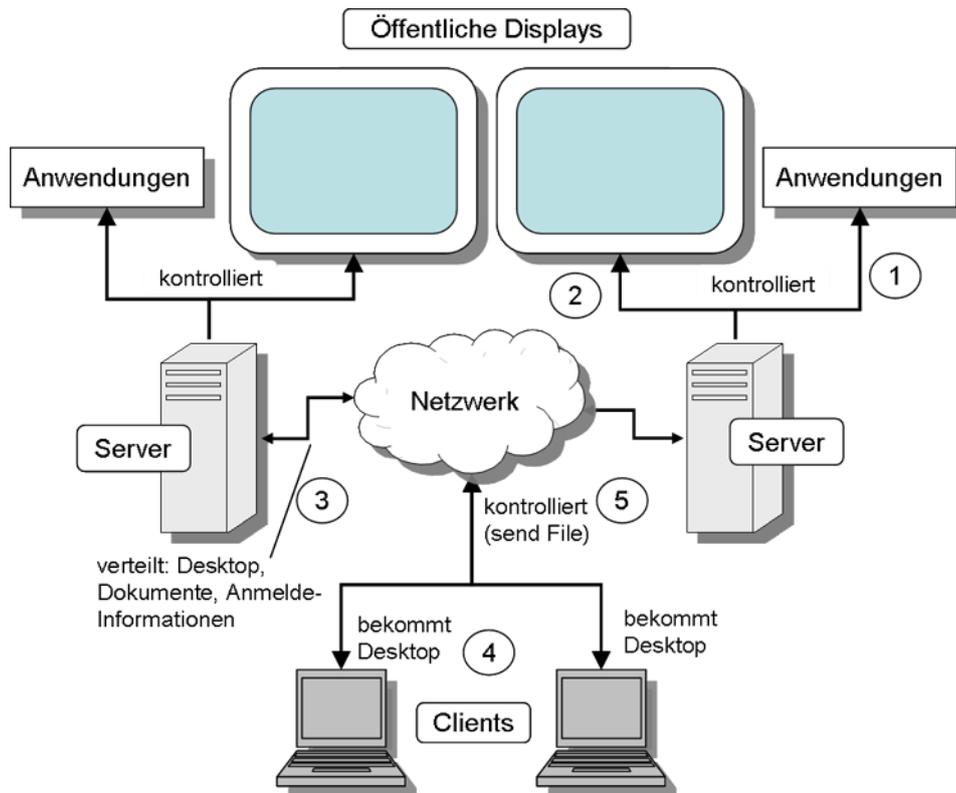


Abbildung 4.2.: Entwurf

Komponenten benutzen das gleiche Datenmodell. (4) Jeder Service kann eine eigene GUI-Komponente mitbringen, welche in der Room-Client-GUI-Komponente automatisch integriert wird.

#### 4.1.2. Room-Client und Room-Server

Die bei Room-Client zugrunde liegende MVC-Architektur ist in Abbildung 4.4 (S.80) dargestellt. Der „Controller“ ist in drei Schichten aufgeteilt, welche, wie die MVC-Komponenten, voneinander entkoppelt (Beobachtungsmuster) sind:

- Anwendung: Ist die Schnittstelle zu den beiden anderen Komponenten Model und View. Sie kontrolliert die verbundene Services, GUI und die Daten.
- Service: Beinhaltet die funktionale Logik des betreffenden Dienstes.
- Kommunikation: Die Kommunikationsschicht, welche den Datenaustausch im Netzwerk kontrolliert.

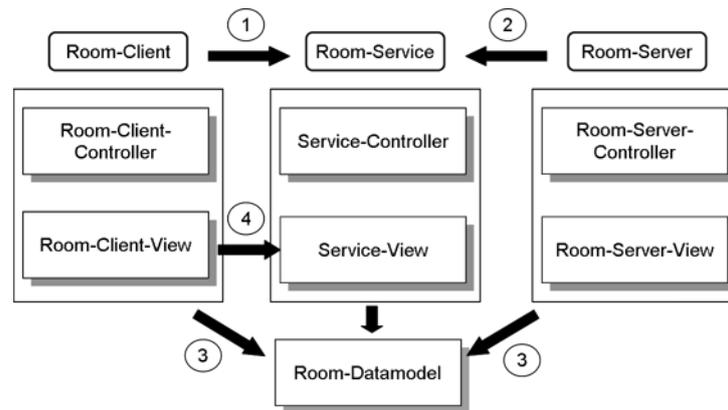


Abbildung 4.3.: Model-View-Controller der einzelnen Komponenten

Die View-Komponente besteht aus der GUI der Hauptanwendung und, wenn vorhanden, der GUI der Services. Die Service-GUI werden über eine speziellen GUI-Komponente, einer wiederverwendbaren Toolbox, in die Haupt-GUI dynamisch integriert. Das Datenmodell besteht aus einem Benutzermodell, welches die angemeldeten Benutzer darstellt und einem Modell für die Dokumente. Die Room-Server-Komponente ist ähnlich aufgebaut, nur fehlt dort eine Service-GUI bei der Präsentationskomponente.

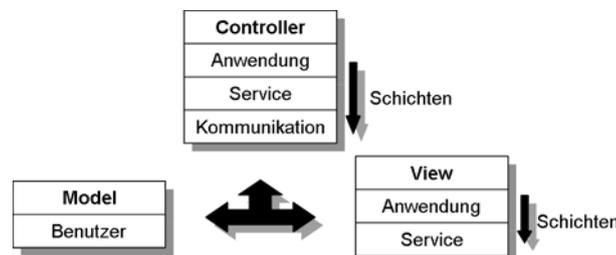


Abbildung 4.4.: Entwurf - Client MVC-Architektur

### 4.1.3. Room-Services

Ein Room-Service läuft unabhängig von dem Room-Server (der Room-Server ist die darüberliegende Anwendungsschicht für den Benutzer) und stellt für die Clients eine bestimmte Funktionalität bereit. Der Room-Service kann nach eigenen Wünschen in die Client-Anwendung eingebunden werden - für die Benutzung des Room-Service stehen Schnittstellen bereit. Der beim Prototyp realisierte Service beinhaltet die Funktionalität gemäß der analysierten Anwendungsfälle, d. h. Anmeldung, Kopieren und Verteilung von Dokumenten.

#### 4.1.4. Kommunikation

Für die Kommunikation wurde die etablierte Middleware „Corba<sup>6</sup>“ benutzt, d.h. sämtliche Kommunikationsschnittstellen für entfernte Aufrufe sind als Corba-Schnittstellen bei den beteiligten Komponenten zu finden und bilden die Kommunikationsschicht in dieser Architektur. Tatsächlich ist ein Service ein Corba-Objekt (Server-Objekt), welcher über die Corba-Schnittstellen angesprochen werden kann. Die Kommunikation von Room-Server zu Room-Client ist über einen Callback-Objekt gelöst, welcher bei dem Anmeldevorgang an den Server übergeben wird (siehe Sequenzdiagramm).

#### 4.1.5. Service-Verwaltung

Als Service-Verwaltung wird der Namensdienst des Corba-Framework benutzt. Die Services werden über den Namensdienst veröffentlicht, d.h. in einen definierten Namensverzeichnis eingetragen. Über die Funktionen des Namensdiensts kann der Client nach bestimmten Service-Objekten suchen und dynamisch Verbindungen aufbauen. Die Service-Verwaltung bildet den in der Abbildung 4.1 (S.78) „Entwurf - Client/Server-Architektur mit Service-Bus/Verwaltung“ dargestellten Service-Bus ab.

#### 4.1.6. Entwurfsentscheidungen

##### 4.1.6.1. Kommunikation

Eine grundlegende Entwurfsentscheidung war die Benutzung des Corba-Frameworks. Mit Corba lassen sich relativ schnell Client/Server-Applikationen entwickeln. Vorteile sind sowohl in der Plattform- und Programmiersprachenunabhängigkeit zu sehen als auch in der bereitgestellten Funktionalität durch die Corba-Dienste<sup>7</sup>. Der Nachteil mag die starke Kopplung zwischen den aufrufenden und aufzurufenden Komponenten sein (wird die Schnittstelle des Corba-Objektes/Klasse geändert, so muss diese Änderung auch bei allen aufrufenden Objekten respektive Klassen vollzogen werden<sup>8</sup>).

---

<sup>6</sup>Common Object Request Broker Architecture

<sup>7</sup>Corba bietet unterschiedliche Dienste an: Name Service (Namensdienst), Event Service (Ereignisdienst), Life Cycle Service (Lebenszyklusdienst), Persistent Object Service (Objekt-Persistenzdienst), Transaction Service (Transaktionsdienst), Concurrency Control Service (Nebenläufigkeitsdienst), Relationship Service (Beziehungsdienst), Externalization Service (Auslagerungsdienst), Security Service (Sicherheitsdienst), und einige mehr

<sup>8</sup>Die starke Kopplung kann durch „Dynamic Invocation Interface“ (DII), „Interface Repository API“ und „Dynamic Skeleton Interface“ (DSI) abgeschwächt oder gänzlich gelöst (kompliziert) werden.

#### 4.1.6.2. Programmiersprache und Plattform

Durch die Benutzung des Corba-Frameworks kämen unterschiedliche Programmiersprachen für den Client und den Server/Service in Frage. Letztendlich wurde Java für beide Komponenten gewählt, da Java bereits auf vielen Plattformen läuft, die Komponenten somit unabhängig von der Plattform<sup>9</sup> sind.

#### 4.1.6.3. Hardware und Einsatz

Room-Client, Room-Service, Room-Service-Verwaltung (Namensdienst) und Room-Server können theoretisch auf unterschiedliche Computer eingesetzt werden. Für den Prototypen wurde die in Abbildung 4.5 (S.82) gezeigte Laufzeit-Konfiguration der einzelnen Komponenten gewählt.

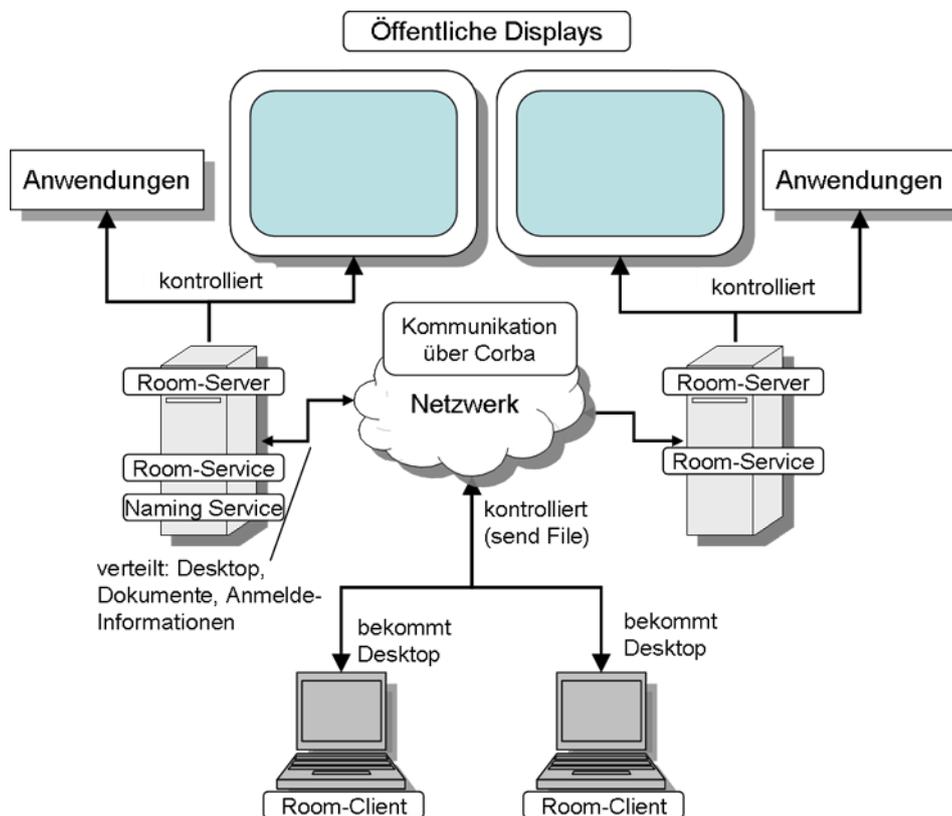


Abbildung 4.5.: Realisierung - Verteilung der Komponenten, Deployment

<sup>9</sup>tatsächlich können die Client- und Server-Komponenten auf unterschiedlichen Plattformen wie WinCE, Linux, Microsoft Windows, Mac OS X etc.. gestartet werden, vorausgesetzt, es existiert dort eine Java-Runtime.

### 4.1.7. Sequenzdiagramme

Die Kommunikation der Komponenten untereinander soll exemplarisch anhand zweier Sequenzdiagramme beschrieben werden:

#### 4.1.7.1. Systemanmeldung und Service-Bereitstellung

In diesem Sequenzdiagramm wird die Service-Bereitstellung durch die Service-Verwaltung (Naming Service) und der gesamte Login-Vorgang gezeigt. In Abbildung 4.6 (S.84) ist zusätzlich die Benachrichtigung andere Clients über den Eintritt des neuen Benutzers in das System zu sehen ((11) - (14), unterhalb des weißen Balkens; ab hier bezieht sich die Sequenz auf andere Client-Objekte). Beschreibung des Sequenzdiagramms:

1. RoomServer erzeugt einen RoomService und
2. registriert diesen beim Naming-Service (Service-Verwaltung).
3. Die Client-Anwendung (ProjektRoomClientApp) fordert von dem Naming-Service sämtliche registrierte Dienste an.
4. Auswahl der gewünschten Dienste durch den Benutzer.
5. Für sämtliche ausgewählte Dienste wird jeweils ein PRClientService-Objekt erzeugt, welches den Dienst kapselt.
6. Für die Kommunikation benötigt der Server/Service ein Client-Kommunikationsendpunkt, das Callback-Objekt „RoomClientImpl“. Das Objekt wird nur einmal für alle Server/Services erzeugt.
7. Hier erfolgt die Kommunikation über das Netzwerk - die Login-Nachricht mit den Namen des Benutzers, Bild (wenn vorhanden) und das Callback-Objekt.
8. ProjectRoomServer und RoomServer sind über das Beobachtungsmuster entkoppelt, daher wird hier ein Ereignis an ProjectRoomServer gesendet.
9. Aktualisierung der GUI (Bild, Name) und Eintrag des Benutzers in das lokale Projektverzeichnis
10. Sollten andere Benutzer angemeldet sein, so werden diese über den neuen Benutzer informiert.
11. Die nachfolgenden involvierten Client-Objekte sind nicht mit den obigen zu verwechseln, sie beziehen sich auf andere angemeldete Benutzer und deren Room-Applikationen.

12. Weiterreichung der LoginNewUser-Nachricht an PRClientService (Beobachtungsmuster)
13. wie 12, nur ist der Empfänger der Nachricht/Ereignis die Hauptapplikation.
14. Aktualisierung der GUI (Anzeige des neuen Benutzers mit Bild und Namen).

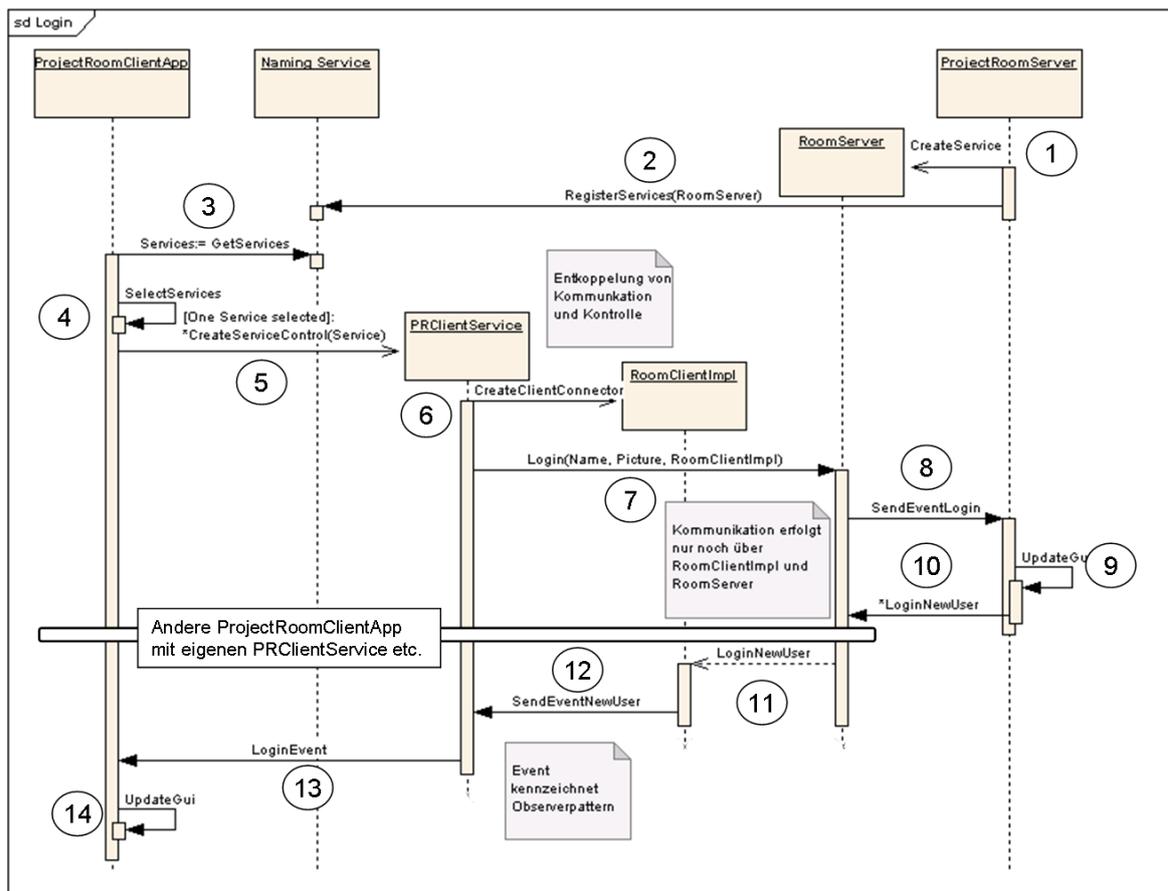


Abbildung 4.6.: Sequenzdiagramm 1 - Service-Bereitstellung und Systemanmeldung

#### 4.1.7.2. Kopieren und Bearbeiten eines Dokumentes

Via Drag-And-Drop wird ein Dokument auf einem entfernten Desktop kopiert und dort mittels einer Anwendung bearbeitet. Dieses Sequenzdiagramm zeigt die beteiligten Objekte beginnend nach der Drag-And-Drop-Aktion:

1. Das Objekt „ProjectRoomClientApp“ gibt das Drag-And-Drop-Dokument an das Service-Objekt weiter. .

2. Das Dokument (jetzt als Object-Byte-Stream mit Dokumentnamen und Name des Benutzers) wird an das entfernte Service-Objekt weitergereicht.
3. Der Service, RoomServer, informiert ProjectRoomServer, der Controller-Instanz der gesamten Anwendung, über das Ereignis „neues Dokument und Bearbeitung“.
4. ProjectRoomServer ruft eine passende im System installierte Anwendung für das Dokument auf und öffnet dieses.

Insgesamt gibt es vier Controller-Instanzen: Die beiden Controller, welche für die gesamte Anwendung zuständig (Client-Anwendung und Serveranwendung) sind und die anderen zwei, verantwortlich für den Service-Bereich (client- und serverseitig). Es wurde versucht eine möglichst schwache Kopplung zwischen Anwendungs- und Service-Controller zu erreichen.

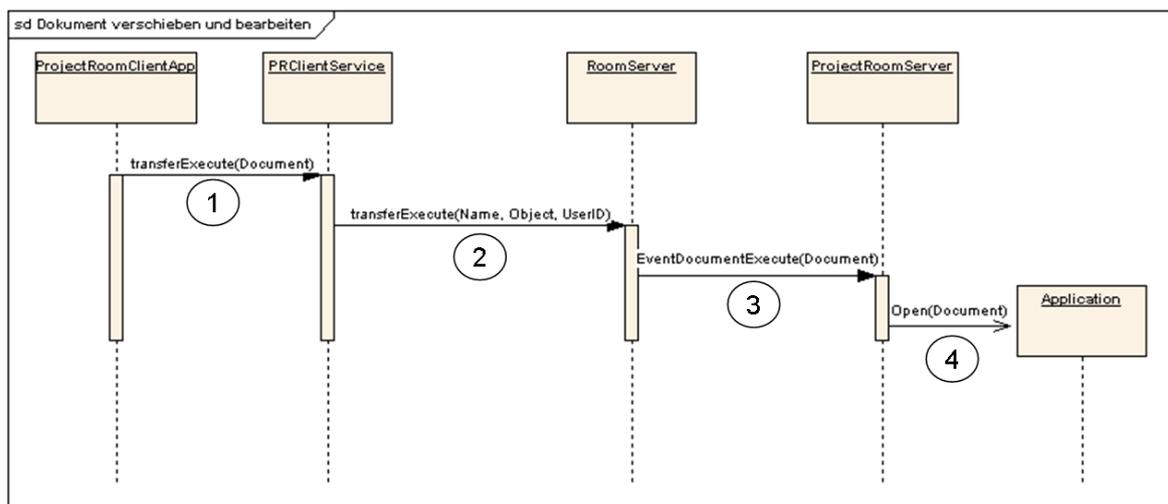


Abbildung 4.7.: Sequenzdiagramm 2 - Kopieren eines Dokumentes auf den Server und anschließender Bearbeitung durch eine Anwendung

## 4.2. Implementierung

Dieses Kapitel zeigt im Wesentlichen die graphische Benutzeroberfläche (GUI) des Prototyps. Der Prototyp teilt sich in zwei Anwendungen auf (unabhängig vom Service).

1. Die Client-Anwendung „ProjectRoomClientApp“
2. Die Server-Anwendung „ProjectRoomServer“

Die zweite Anwendung bietet, wie die Client-Anwendung, eine Drag-And-Drop-Funktionalität, welche für die Verteilung von Dokumenten benötigt wird. Gedacht ist dieses für den Fall, dass ein Vortragender (stehend vor einem großen interaktiven Display, z. B. eine Powerwall), wie auch die Zuhörer (die anderen Projektteilnehmer), Dokumente verteilen möchte. Der Einfachheit halber wurde in dieser Anwendung auch die Service-Funktionalität eingebunden sowie eine Konfigurationsmöglichkeit (siehe Anwendungsfall A.5 S.111 „Administration des Servers“) des Servers/Services realisiert. Die in diesem Kapitel gezeigten Abbildungen der beiden Applikationen sind vergrößert im Anhang B S.112 zu finden.

## 4.2.1. ProjectRoomClientApp

### 4.2.1.1. Initialisierte Anwendung

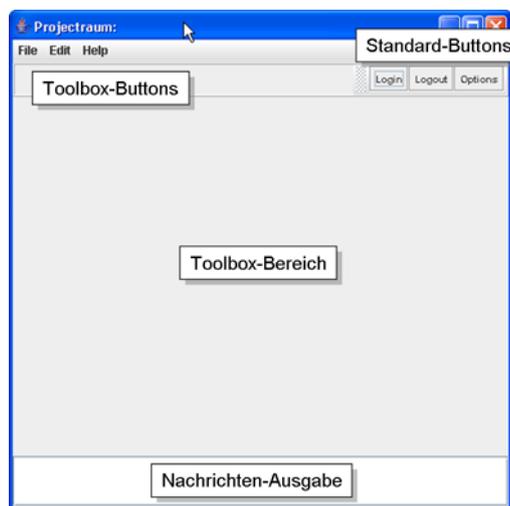


Abbildung 4.8.: ProjectRoomClientApp - Gestartete Anwendung

Die Abbildung 4.8/B.1 zeigt die Anwendung kurz nach dem Start. Die einzelnen visuellen Bereiche teilen sich auf in:

- **Toolbox-Bereich:** Hier erscheinen die dem Service zugehörigen Toolboxes. Jeder Service hat die Möglichkeit seine eigene GUI in der ProjectRoomClientApp-GUI zu integrieren. Die Service-GUI kann aus insgesamt drei Teilen bestehen:
  - Einer Haupt-GUI, die in Zentrum dargestellt wird, und
  - zwei weiteren, welche links und recht der Haupt-GUI angeordnet sind.

Der mittlere Bereich, d. h. die Anzeige der Haupt-GUI, dient gleichzeitig der Auswahl eines Services. Die Haupt-GUIs der Services teilen sich diesen Bereich und werden automatisch gekachelt dargestellt (siehe Abbildung 4.9/B.2)

- **Standard-Buttons:** Diese Buttons<sup>10</sup> gehören fest zu der Hauptanwendung. Mit den Buttons gelangt der Anwender zu den Dialog-Boxen: Login für die Anmeldung, Logout für die Abmeldung, Options für die Konfiguration der Anwendung.
- **Toolbox-Buttons:** Jeder Service hat die Möglichkeit seine eigenen GUI-Elemente in die Hauptanwendung zu integrieren (automatisch). Spezielle Service-Buttons erscheinen in dieser Leiste.
- **Nachrichten-Ausgabe:** Diese Fläche dient der Nachrichtenausgabe der Anwendung. Hier bekommt der Anwender beispielsweise aussagekräftige Rückmeldungen bei der Anmeldung.
- **Standard-Menü:** Herkömmliches Menü u. a. zum Beenden der Anwendung.

#### 4.2.1.2. Anwendung in angemeldetem Zustand

Hat sich der Benutzer angemeldet, so füllt sich der Toolbox-Bereich mit unterschiedlichen Toolboxes (immer abhängig vom Service) - siehe Abbildung 4.9/B.2:

- **Remote Desktop:** In der Mitte sind die zwei Toolboxes der „entfernten Desktops“ zu sehen. Die Toolbox-Inhalte werden ständig mit den aktuellen Desktops aktualisiert. Diese Toolboxes ermöglichen nicht nur die Kontrolle der „entfernten Desktops“, sondern dienen auch als Schalter für die beiden anderen Toolboxes (Dokumente, Benutzer). Je nach selektierter Toolbox ändert sich der Inhalt der beiden anderen (bei der Powerwall mögen sich beispielsweise andere Benutzer angemeldet haben, die aktuell nicht zu sehen sind). Die anderen Toolboxes sind somit kontextsensitiv, abhängig vom gewählten „entfernten Desktop“. Diese Toolbox stellt die Funktionalität gemäß Anwendungsfall A.4 bereit: Über Drag-And-Drop können Dokumente auf die „entfernten Desktops“ verschoben werden. Dort werden diese dann in einer passenden Anwendung automatisch geöffnet.
- **Dokumente:** Hier liegen die Dokumente für einen entfernten Desktop. Der Anwender kann die Dokumente (Artefakte) via Drag-And-Drop (die Dokumente werden in diese Toolbox fallen gelassen) auf den Server/Service transferieren - die Dokumente werden dort in einer speziellen Toolbox strukturiert für andere Benutzer dargestellt (z. B. dem Vortragenden).

---

<sup>10</sup>button = Knopf, grafische Schaltfläche in der GUI-Gestaltung

- **Benutzer:** Andere beim Service (hier Touchscreen oder Powerwall) angemeldete Benutzer werden hier angezeigt. Hier können Dokumente mittels Drag-And-Drop an die anderen Benutzer (Abbildung der Benutzer) verteilt werden.

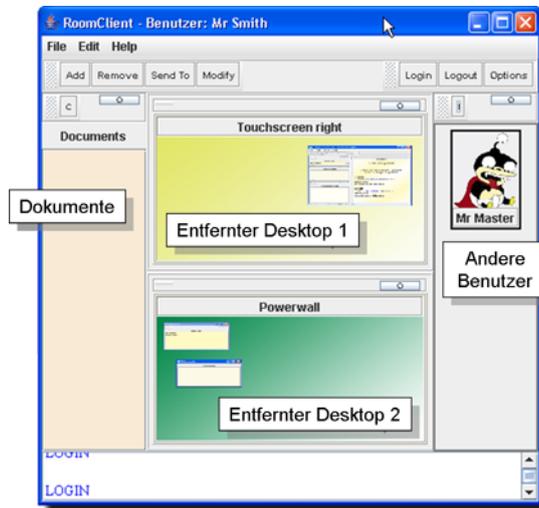


Abbildung 4.9.: ProjectRoomClientApp - Der Benutzer hat sich angemeldet

#### 4.2.1.3. Toolbox

Die oben beschriebenen Toolboxen sind als Behälter für andere GUI-Elemente anzusehen. Ihre Besonderheit liegt in der erweiterbaren Toolbox-Button-Leiste sowie in den umschaltbaren Darstellungsmodi:

- **Modus 1:** Ist der „embedded“-Mode (eingebetteter Modus), die GUI des Services wird innerhalb der Haupt-GUI dargestellt.
- **Modus 2:** Ist der „free“-Mode (freier Modus), die GUI des Services wird in einem frei verschieb- und skalierbaren Fenster dargestellt.

Die Umschaltung erfolgt über den in der Abbildung 4.10/B.3 gekennzeichneten Modus-Button. Der zum Service gehörende Button schaltet von der grafischen (Abbildung und Name) zur rein textuellen (nur Name) Anzeige der angemeldeten Benutzer um.

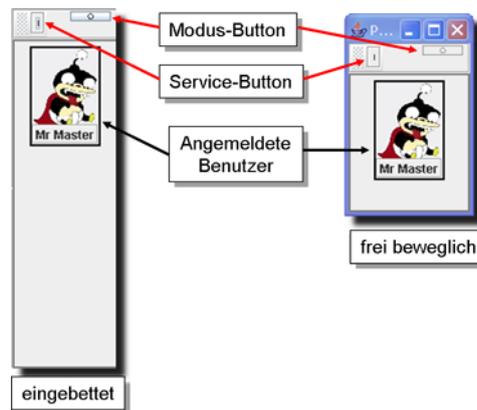


Abbildung 4.10.: Toolbox - Freier und eingebetteter Modus

#### 4.2.1.4. Verteilte Toolboxes

Die einzelnen Toolboxes lassen sich frei auf dem lokalen Desktop anordnen. In Abbildung 4.11/B.4 sind die Toolboxes für die beiden „entfernten Desktops“ nach der tatsächlichen räumlichen Anordnung der Powerwall und Touchscreen ausgerichtet.

### 4.2.2. ProjectRoomServer

#### 4.2.2.1. Initialisierte Anwendung

Die Abbildung 4.12/B.5 zeigt die Server-Anwendung des Prototypen. Der eigentliche Service wurde noch nicht registriert respektive der Server nicht gestartet. Die wichtigsten GUI-Komponenten sind:

- **Benutzer-Toolbox:** Die angemeldeten Benutzer werden in dieser Toolbox angezeigt. Wie aus der ProjectRoomClientApp bekannt, kann auch hier zwischen einer textuellen (Namen der Benutzer) oder grafischen Darstellung (Abbildungen und Namen der Benutzer) umgeschaltet werden. Sie dient dem Verteilen von Dokumenten an angemeldete Benutzer (Drag-And-Drop).
- **Dokumente-Toolbox:** In dieser Toolbox werden die Dokumente (Artefakte) der einzelnen angemeldeten Benutzer in einer Baumstruktur angezeigt. Die obersten Elemente sind die Namen der Benutzer, darunter befinden sich die einzelnen Artefakte. Der Behälter dient der Organisation der aktuell von den einzelnen Benutzern zur Verfügung gestellten Dokumente.

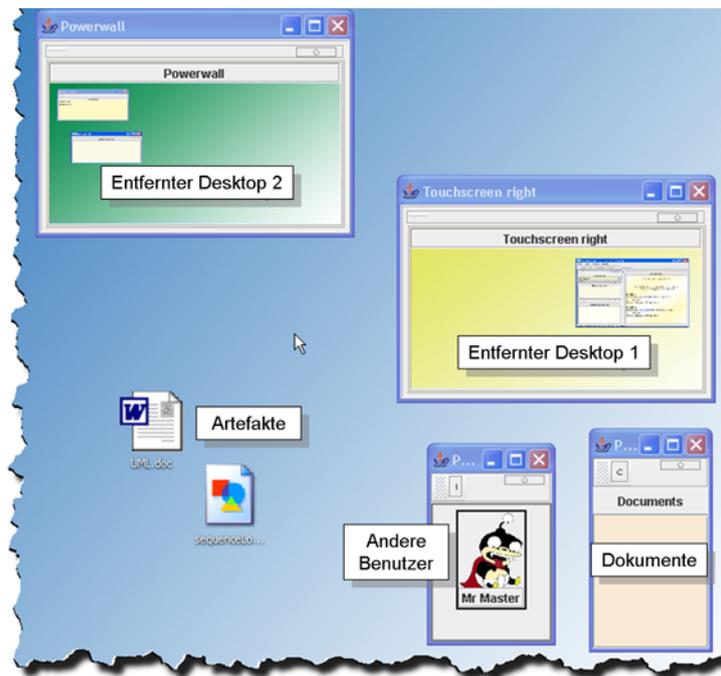


Abbildung 4.11.: ProjectRoomClientApp - Die Toolboxen liegen (frei verschiebbar) auf dem lokalen Desktop

- **Anwendungen-Toolbox:** Die bei der Administration eingetragenen gültigen Anwendungen sind in dieser Toolbox zu sehen. Sollen Dokumente entfernt in einer auf dem Server-System laufende Anwendung geöffnet werden, so muss dem System mitgeteilt werden, welche Anwendung für welchen Dokumenten-Typ verantwortlich ist (siehe Unterkapitel: 4.2.4.2 „Administration von ProjectRoomServer“).
- **Ausgabe-Toolbox:** Informationsanzeige des Servers. Sämtliche Aktionen (Anmeldung, Dokumente verschieben/verteilen, etc.) die den Server betreffen, werden protokolliert und hier angezeigt.
- **Toolbar:** Die Toolbar stellt Buttons (Schalter) bereit, die es dem Anwender (hiermit ist der Anwender gemeint, der direkt vor dem Ein- / Ausgabegerät, z.B. der Powerwall, steht) ermöglichen:
  1. den Server/Service zu starten oder anzuhalten (Start/Stop-Buttons),
  2. Angemeldete Benutzer serverseitig abzumelden respektive zu entfernen („Benutzer entfernen“-Button),
  3. und den Server zu konfigurieren bzw. zu administrieren (Options-Button)

- **Statusleiste:** Zeigt den aktuellen Status des Servers an (Server gestartet, beendet; Verbindungsinformationen).

Die einzelnen Toolboxen sind über GUI-Raumteiler getrennt, die horizontale und vertikale Anpassungen der Toolbox-Größen sowie deren Minimierung (über die kleinen Pfeile) erlauben.

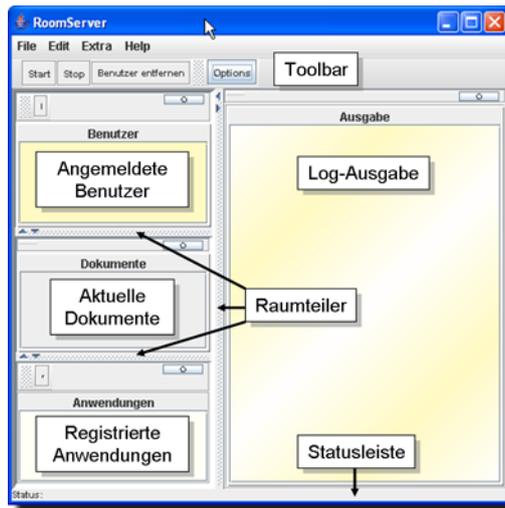


Abbildung 4.12.: ProjectRoomServer - Gestartete Anwendung

#### 4.2.2.2. Server-Anwendung - Service gestartet

Ist nicht nur die Anwendung, sondern auch der Service (über den „Start-Button“) gestartet, so wird dieses, wie in Abbildung 4.13/B.6 zu sehen ist, in der GUI „widergespiegelt“. Die Ausgabe berichtet über den erfolgreichen Start des Services und zusätzlich über die erfolgreiche Anmeldung eines neuen Benutzers. Der angemeldete Benutzer ist mit einer Abbildung in der Benutzer-Toolbox zu sehen. Die Statusleiste zeigt eine erfolgreiche Verbindung zur Service-Verwaltung (Name des Nameservers, IP-Adresse<sup>11</sup> und Verbindungsport) sowie den Status des Services („running“, am Laufen) an.

<sup>11</sup> Internet-Protokoll-Adresse

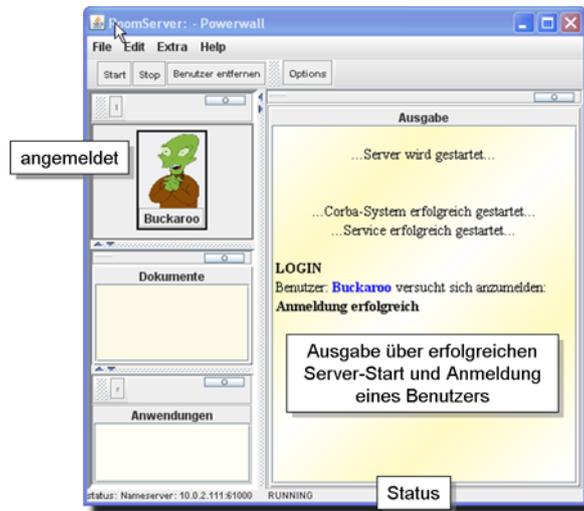


Abbildung 4.13.: ProjectRoomServer - Gestartete Anwendung und Service

#### 4.2.2.3. Verteilte Toolboxes beim Server

Wie bei ProjectRoomClientApp können auch hier die Toolboxes aus der Haupt-GUI herausgelöst und auf dem Desktop verteilt werden. Die Toolboxes können zusätzlich, je nach Belieben, vergrößert oder verkleinert werden. Die Abbildung 4.14/B.7 zeigt in einem Ausschnitt des Server-Desktop (Powerwall) eine Beispielanordnung der beiden Toolboxes „Documents“ und „Benutzer“ - die Server-Ausgabe wurde nicht „herausgelöst“.



Abbildung 4.14.: ProjectRoomServer - Verteilte Toolboxes auf dem Server-Desktop

### 4.2.3. Interaktionen

In diesem Abschnitt sind die Interaktionen mit den einzelnen GUI-Elementen beschrieben. Die beiden Hauptteile des Prototyps, „ProjectRoomClientApp“ und „ProjectRoomServer“, ermöglichen die in den Anwendungsfällen 1-3 (Anhang A, S.107 - 109) beschriebenen Interaktionen mittels graphischer Benutzerschnittstelle (GUI).

#### 4.2.3.1. Benutzer-Anmeldung

Die Anmeldung eines neuen Benutzers soll sowohl beim Server sichtbar sein als auch bei anderen, bereits angemeldeten Benutzern. Beide Anwendungen („ProjectRoomClientApp“ und „ProjectRoomServer“) besitzen für die Darstellung der (anderen) angemeldeten Benutzer die bereits erwähnte Toolbox. Die Anmeldung wird über die Anmelde-Dialogbox vollzogen - zum Öffnen des Dialogs muss der Login-Button in der Toolbar gedrückt werden (siehe Abbildung 4.8/B.1). Abbildung 4.15/B.8 zeigt die einzelnen Schritte der Anmeldung:

1. **Server-/Service-Auswahl:** Hier sind aus einer Liste die Server/Services auszuwählen.
2. **Benutzerangaben:** Als nächstes erfolgt die Eingabe von Benutzername und Passwort.
3. **Auswahl der Abbildung:** Optional kann ein Bild (Avatar) ausgewählt werden, welches der grafischen Benutzeridentifikation im CSCW-Raum dient. Der Benutzer kann ein Bild aus einer Bildersammlung (Avatar-Auswahl-Dialog) wählen oder in der Konfiguration sein eigenes angeben (z. B. Passfoto).
4. **Abschließen:** Der Login-Vorgang muss noch durch den „Login-Button“ abgeschlossen werden.

Die Benutzer-Toolbox sämtlicher angemeldeter Benutzer und der betreffende Server werden automatisch mit dem neuen Benutzer (Name und gegebenenfalls Bild) aktualisiert. Die Benutzer-Toolbox ist, wie wir später noch sehen werden, speziell für die Verteilungsinteraktion konzipiert. Abbildung 4.16/B.9 zeigt die Anmeldung dreier Benutzer an einem Server und den Inhalt der Toolboxes aus Sicht des Servers und des Benutzers „Buckaroo“.

#### 4.2.3.2. Dokument transferieren und Initiierung eines Bearbeitungsprozesses

Der angemeldete Benutzer möchte ein Artefakt auf einem entfernten Desktop öffnen. In diesem konkreten Beispiel soll eine Bild-Datei in PNG<sup>12</sup>-Format innerhalb des Desktops der „Po-

<sup>12</sup>Portable Network Graphic, Format zur Speicherung von Bitmap-Grafiken

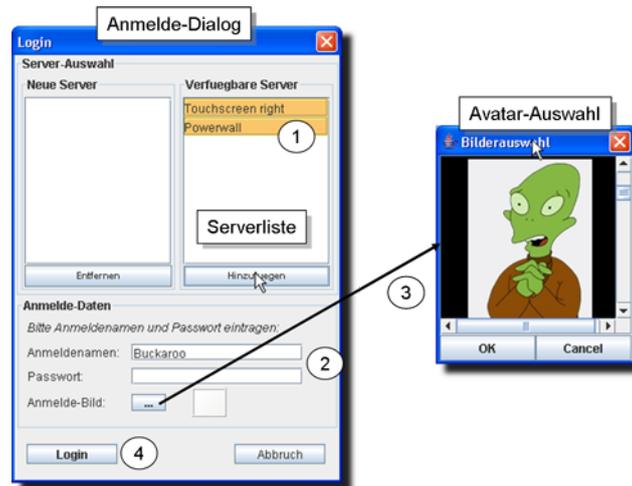


Abbildung 4.15.: Benutzer-Anmeldung - Dialog zur Anmeldung

werwall“ angezeigt werden. Die Abbildung 4.17/B.10 zeigt einen Ausschnitt des „Powerwall-Desktops“ vor dem Kopieren. Bisher hat sich nur ein Benutzer („Buckaroo“) angemeldet, welcher - zu sehen an der leeren Documente-Toolbox - noch keine Dokumente für den Server zur Verfügung gestellt hat. Der Desktop des Benutzers und der Verschiebe/Kopier-Vorgang ist in der Abbildung 4.18/B.11 zu sehen. Der angemeldete Benutzer verschiebt (tatsächlich kopiert der Benutzer) das Bild „sequenceLogin4.png“ auf den entfernten Desktop der „Powerwall“ (Powerwall-Toolbox) via Drag-And-Drop. Wie die Dokumente-Toolbox der „Powerwall“ ist auch die eigene Toolbox zu diesem Zeitpunkt leer. Nachdem der Benutzer das Bild über die Powerwall-Toolbox fallen gelassen hat, erscheint dieses, geöffnet in einer adäquaten Anwendung, auf dem Desktop der „Powerwall“ (Abbildung: 4.19/B.12). Gleichzeitig ist dort das Bild respektive das Artefakt mit Namen in der Dokumente-Toolbox zu sehen - eingeordnet unter dem Benutzernamen. Als Bildbetrachter wurde vorher das Programm „IrfanView<sup>13</sup>“ beim Server registriert. Das Artefakt ist gleichzeitig in die Dokumente-Toolbox (Dokumente-Behälter für eine Sitzung) des Benutzers kopiert wurden. (siehe Abbildung 4.20/B.13).

#### 4.2.3.3. Verteilen von Dokumenten

Das Verteilen von Dokumenten gestaltet sich ähnlich wie bei „Dokument transferieren und Initiierung eines Bearbeitungsprozesses“. Abbildung 4.21/B.14 zeigt die beiden Desktops der angemeldeten Benutzer „Mr Master“ und „Buckaroo“ - der dritte Benutzer, „Ernie“, ist in der nachfolgenden Aktion nicht involviert. Zu beachten ist hierbei, dass die Dokumente-Toolbox von „Buckaroo“ leer ist. Auch hier kommt die Drag-And-Drop-Interaktion zum Einsatz: Auf

<sup>13</sup>IrfanView ist ein Freeware-Bildbetrachter, welcher unterschiedliche Grafik-Formate unterstützt. Irfanview ist nur für den privaten Gebrauch und den Einsatz in Ausbildungseinrichtungen frei.

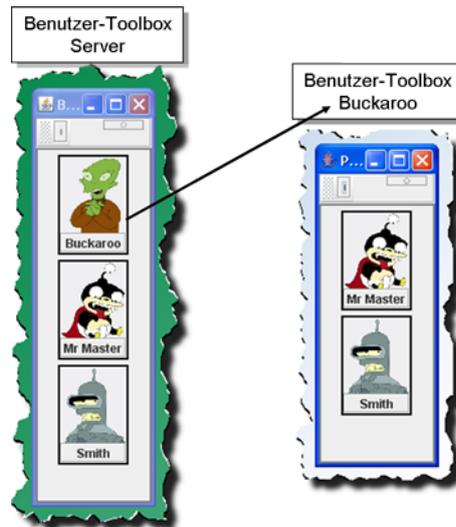


Abbildung 4.16.: Benutzer-Anmeldung - Inhalt der Benutzer-Toolboxen

den Desktop liegende Artefakte werden auf die in der Benutzer-Toolbox angezeigte Abbildung des Ziel-Benutzers (Buckaroo) „fallen gelassen“. Der von der Verteilung betroffene Benutzer erhält eine Kopie des Artefaktes in seiner Dokumente-Toolbox (siehe Abbildung 4.22/B.15)

#### 4.2.4. Konfiguration/Administration der Anwendungen

Hier wird nur kurz, der Vollständigkeit halber, auf die Konfigurations- respektive Administrationsmöglichkeiten der beiden Anwendungen eingegangen.

##### 4.2.4.1. Konfiguration von ProjectRoomClientApp

Die Konfiguration von ProjectRoomClientApp besteht aus insgesamt zwei Seiten, angezeigt als auswählbare Register (siehe Abbildung 4.23/B.16/B.17).

1. **System:** Hier können die Verbindungsdaten für den Name-Server (hier die Service-Verwaltung) angegeben werden. Die Verbindungsdaten bestehen aus: IP-Adresse, Subnet-Mask sowie die Portnummer. Die Verbindung zur Service-Verwaltung kann über den Button „Verbindungstest“ getestet werden.
2. **Benutzer:** Speicherort der Dokumente (für die Dokumente in der Dokumente-Toolbox) und ein eigenes Bildverzeichnis lassen sich hier eintragen.

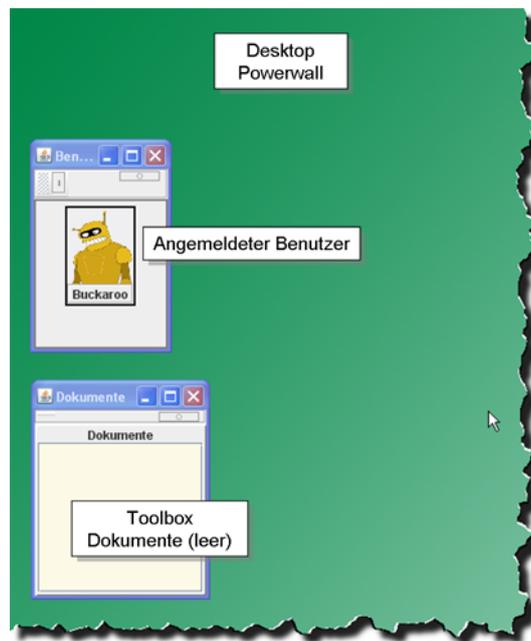


Abbildung 4.17.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop der Powerwall vor der Ausführung

#### 4.2.4.2. Administration von ProjectRoomServer

Die Administrationsdialogbox ist über den „Options“-Button in der Haupt-GUI zu erreichen. Hier stehen dem Administrator vier Hauptseiten (als Register) zur Verfügung (siehe Abbildung 4.24):

1. **System:** Wie bei ProjectRoomClientApp müssen auch hier die Verbindungsdaten eingegeben werden (große Abbildung B.18).
2. **Anwendung:** Hier werden die registrierten Anwendungen verwaltet (große Abbildung B.19). Sollen neue Anwendungen hinzugefügt werden, so muss die Dialogbox „Neue Anwendung“ über den „Hinzufügen“-Button aufgerufen werden (große Abbildung B.22). Dort lassen sich unterschiedliche Angaben über die neue Anwendung machen - wichtig hierbei ist die Angabe der zugehörigen Datei-Typen.
3. **Desktop:** Diese Einstellmöglichkeiten haben einen erheblichen Einfluss auf den CPU-Ressourcen-Verbrauch. Hier kann der Kompressionsgrad der zu übertragende Desktop-Bilder, die Auflösung und das Übertragungsintervall (fps = frames per second) eingestellt werden. (große Abbildung B.20)
4. **Benutzer:** Angabe des Speicherortes für die Dokumente (große Abbildung B.21)

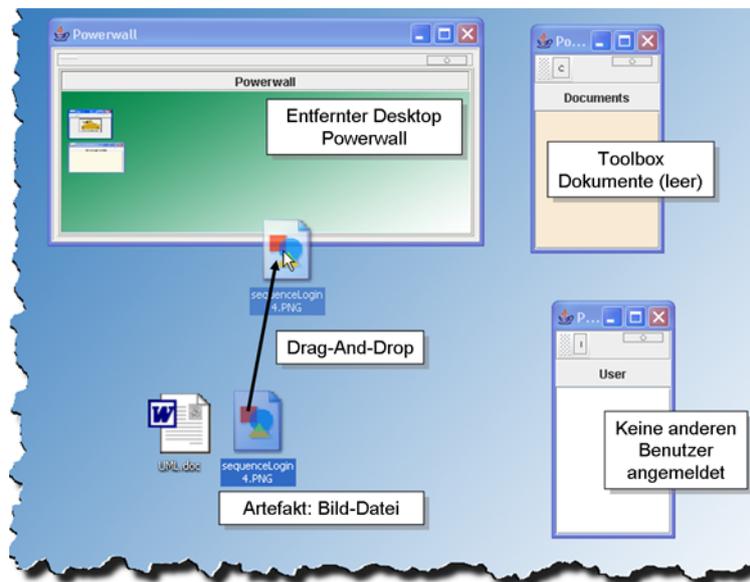


Abbildung 4.18.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers und der Verschiebevorgang

### 4.3. Software-Technisches Fazit

Der Prototyp läuft, bedingt durch den Einsatz der Programmiersprache „Java“ und Corba als Kommunikations-Middleware, auf einer Vielzahl von unterschiedlichen Plattformen (Hardware-Systeme, Betriebssysteme). Wie in dem Unterabschnitt 3.1 (S.54) „Rahmenbedingungen und Abgrenzung“ bereits beschrieben, ist der Einsatz (speziell der Client-Anwendung) im Wesentlichen durch die Größe der Anzeigefläche des benutzten Ein- und Ausgabegerätes begrenzt (siehe Gesamt-Fazit Kapitel 5 S.102). Die beiden Anwendungen, aus denen der Prototyp besteht, sind in einer MVC-Architektur entworfen, und bilden dadurch ein leicht erweiterbares, wiederverwendbares und robustes Gesamtsystem. Der Nachteil dieser Architektur liegt in der Service-Schicht. Die Kommunikationsendpunkte eines Services sind noch zu stark gekoppelt, d. h. eine Änderung der Server-Seite eines Services würde eine Änderung der Client-Seite nach sich ziehen. Der Namensdienst, hier die Service-Verwaltung, dient bisher nur der Service-Auffindung - eine umfassende Verwaltung bietet gewiss noch andere Funktionen (z. B. eine Service-Orchestration, mit dessen Hilfe Services untereinander kombiniert und verbunden werden können - z. B. MDS mit einem Rechteverwaltungsservice). Problematisch ist auch der Entwurf der Server-Anwendung, da diese gleichzeitig für die Service-Bereitstellung und der Benutzerinteraktion mit dem Server (Bereitstellung einer graphischen Benutzerschnittstelle) verantwortlich ist - hier müsste eine klare Trennung bestehen.

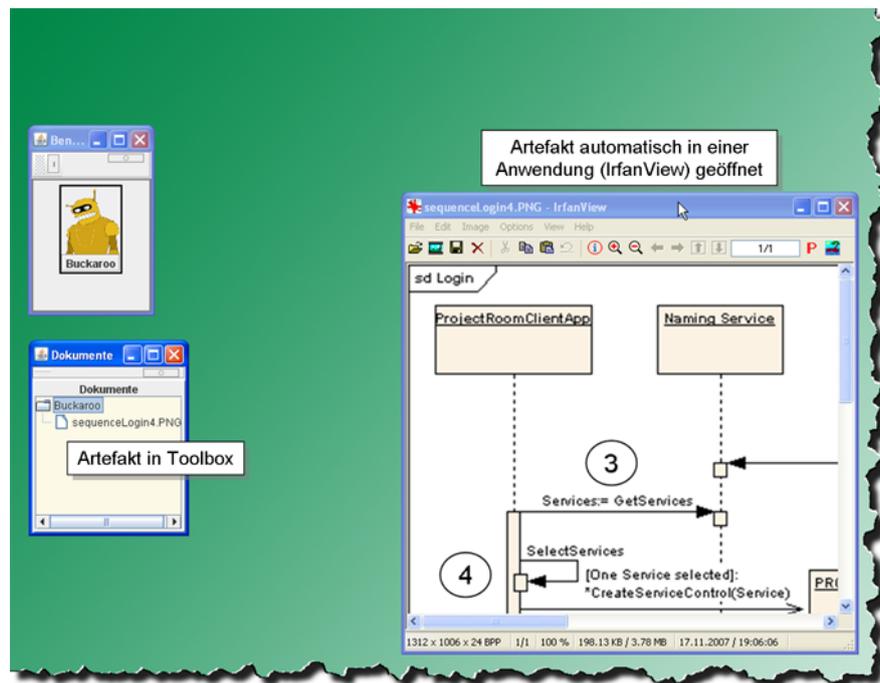


Abbildung 4.19.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses  
- Anzeige des Bildes mit einer Anwendung (IrfanView), aktualisierte  
Dokumente-Toolbox

Eine wesentlich elegantere Architektur für den Prototyp, wäre eine „Service Oriented Architecture“ (SOA) gewesen, mit den drei Instanzen: Service-Konsument, Service-Anbieter und Service-Broker. Ein CSCW-System zeichnet sich u. a. durch den Einsatz von heterogenen Geräten aus, welche eigene Dienste in das Gesamtsystem dynamisch mit einbringen können - hier wäre eine SOA mit einem Enterprise Service Bus (ESB) denkbar. Zu den Aufgaben des Busses gehört die gesteuerte, zuverlässige, nachrichtenbasierte Kommunikation über Systemgrenzen hinweg. Die Services können unabhängig von der zugrunde liegenden Implementierungen über Schnittstellen aufgerufen werden, deren Spezifikationen öffentlich und damit vertrauenswürdig sein können. Die Umsetzung eines SOAs via Web Services auf der Basis bestehender Standards wie SOAP, WSDL, UDDI, ist nicht zwingend - tatsächlich kann prinzipiell jede dienstbasierte Technologie wie DCOM, EJB und auch Corba dafür eingesetzt werden. Eine reine SOA-Lösung mittels Corba ist aber schwierig umzusetzen (beispielsweise gestaltet sich der asynchrone entfernte Methodenaufruf eines Objektes bei Corba schwieriger als bei den Web Services).

Die Vorteile bei einer reinen Corba-Lösung (für die Bereitstellung von Diensten und Kommunikationsinfrastruktur) sind in den unterschiedlichen vorhandenen Diensten (siehe Unterabschnitt 4.1.6.1), die Corba von Hause aus mit sich bringt, und der sicheren Kommunikationsinfrastruktur zu sehen.

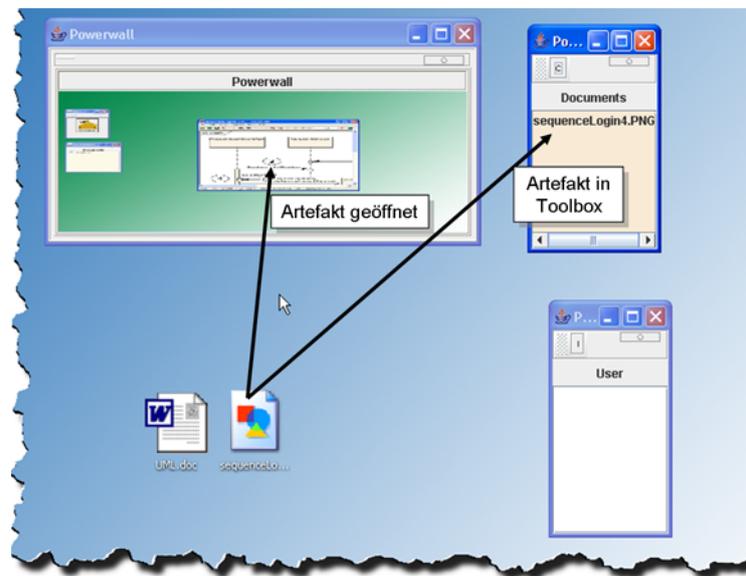


Abbildung 4.20.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers nach der Aktion

Für die Aufgabe des Prototyps, die analysierten Grundfunktionen und die zugehörigen Interaktionen zu testen, ist die Architektur respektive Entwurf und Realisierung mehr als ausreichend. Für ein Gesamtkonzept eines CSCW-Systems sind die Architekturen von Roman Bartnik (Bartnik:2006) oder Peter Tandler (PT:2004) vorzuziehen.

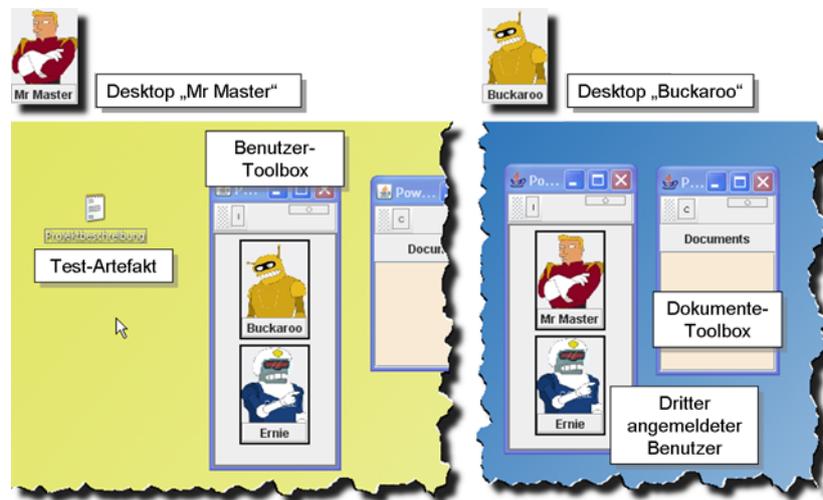


Abbildung 4.21.: Verteilen von Dokumenten - Die Desktops der beiden Benutzer vor der Verteilung

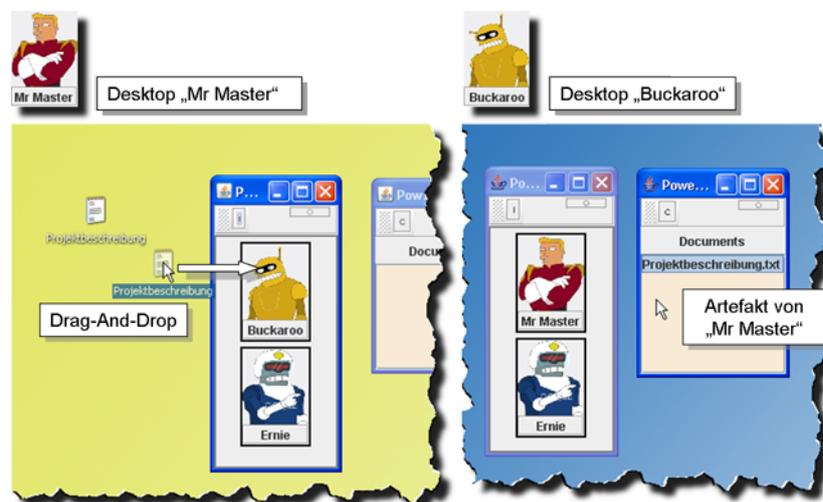


Abbildung 4.22.: Verteilen von Dokumenten - Das Dokument taucht in der Dokumente-Toolbox von „Buckaroo“ auf.



Abbildung 4.23.: Konfiguration von ProjectRoomClientApp - NameSerververbindungs- und Benutzereinstellungen

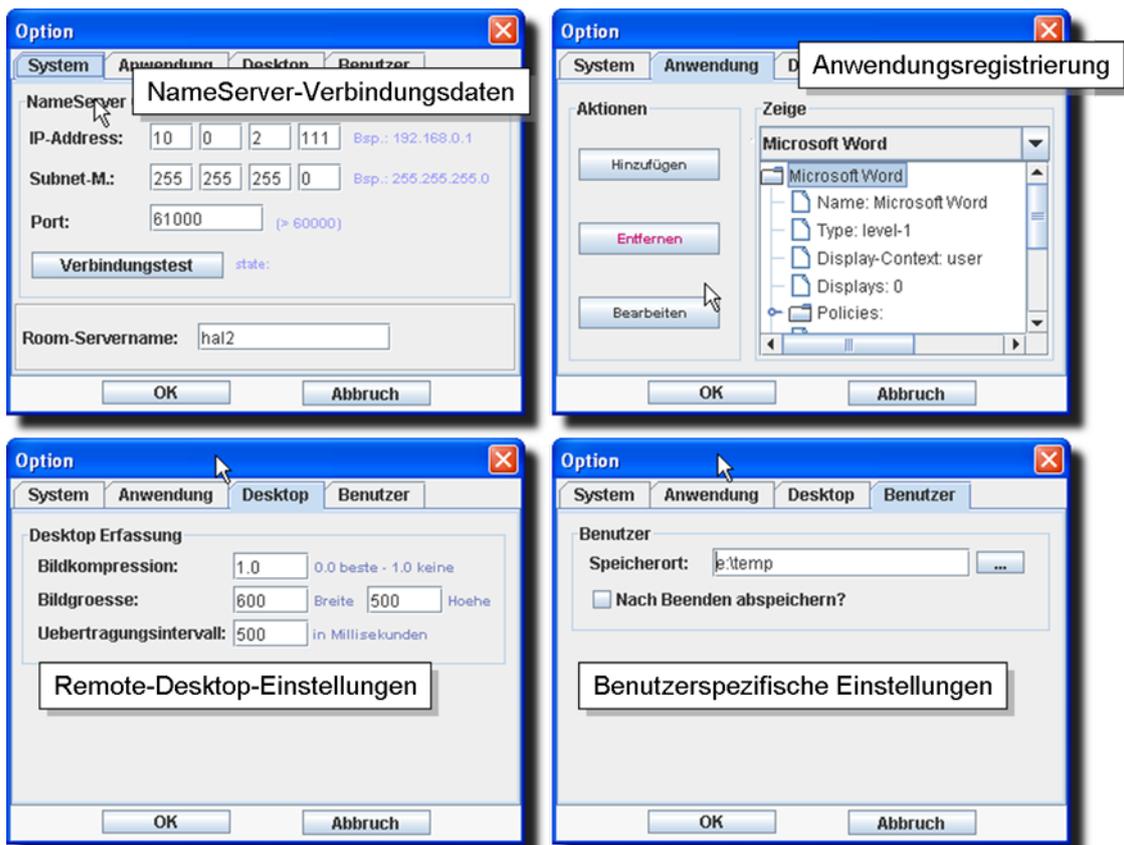


Abbildung 4.24.: Adminstration von ProjectRoomServer - Sämtliche Dialog-Seiten

## 5. Fazit

Die in der Einleitung gesetzten Ziele wurden weitestgehend erreicht:

1. Ausgehend von der Analyse möglicher Szenarien in einem kooperativen Konferenzraum ist eine Grundfunktionalität für ein CSCW-System erarbeitet wurden.
2. Für die Umsetzung der Grundfunktionalität wurden dann mögliche, zur einer CSCW-Umgebung passenden, Interaktionsformen untersucht.
3. Diese wurden dann in einem ersten Prototypen realisiert.

Die Architektur, der Entwurf und die software-technische Realisierung ist bereits in dem vorigen Kapitel 4.3 einer kurzen Kritik unterzogen worden. Hier wird das Gesamtkonzept der Multi-Desktop-Steuerung genauer betrachtet und anhand des Prototyps erste Vor- und Nachteile beschrieben. Die software-ergonomischen Prinzipien für CSCW-Systeme (Kapitel 2.4.3) ermöglichen hierbei eine erste Kritik. Hier sei nochmals darauf hingewiesen, dass die nachfolgende Beurteilung einen noch subjektiven Charakter hat, einen ersten persönlichen Eindruck wiedergibt. Eine Beurteilung der Software-Ergonomie anhand von Fragebogenverfahren, Expertenurteil und Usability Testing wäre der nächste Schritt.

### 5.1. Beurteilung des Prototyps

Das Konzept der Multi-Desktop-Steuerung konnte beim Prototyp umgesetzt werden: Fremd-Desktops werden in den lokalen Benutzer-Desktops integriert und können, in einem gewissen Umfang, gesteuert werden. Auch ist die Grundfunktionalität (An-/Abmeldung, Dokumentenverteilung, Initiierung eines Bearbeitungsprozess) beim Prototyp wiederzufinden. In Bezug auf die in Kapitel 2.4.3 „CSCW-Systeme und Software-Ergonomie“ erwähnten Kriterien lässt sich für den Prototyp „ProjectRoom“ folgendes anmerken:

- **Mentales Modell, Konsistenz und Erwartungskonformität:**

Das mentale Modell eines erweiterten virtuellen Arbeitsplatzes ist durch die flexible Verteilung der „entfernten Desktops“ auf den Client-Desktops und dessen möglichen

Ausrichtungen zu den realen Ein-/Ausgabegeräten in der Client-Anwendung (Room-ClientApp) konzeptuell umgesetzt worden. Der Benutzer kann die einzelnen entfernten Desktops auf dem eigenen Desktop leicht erkennen. Problematisch ist hier nur die Neu-Positionierung des Benutzers mit seinem Ein-/Ausgabe-Gerät (z. B. Laptop) in dem CSCW-Raum: Die individuellen positionierten und zum Original-Desktop ausgerichteten „Toolboxen“ für die „entfernten Desktops“ müssten gegebenenfalls auf dem lokalen Desktop neu angeordnet werden. Dadurch würde Zeit, die eigentlich für die primäre Aufgabe genutzt werden sollte, der sekundären Aufgabe (siehe Unterabschnitt 2.4.3.1 S.35) - Neu-Anordnung der „entfernten Desktops“ - zufallen.

Durch die bekannte und auch hier eingesetzte Drag-And-Drop-Methode zum Verschieben/Kopieren von Dokumenten ist die Ansteuerung des „entfernten Desktops“ und die Verteilung von Dokumenten für den Benutzer intuitiv - die externe Konsistenz, d. h., Erfahrungen und gewohnte Interaktionen aus anderen Systemen wurden berücksichtigt. Insofern verhält sich der Prototyp für den Benutzer erwartungskonform.

- **Kontrolle: Transparenz, Vorhersehbarkeit, Beeinflussbarkeit**

Die Forderungen nach Transparenz und Vorhersehbarkeit, den ersten beiden Stufen des Kontroll-Modells, wurden beim Prototyp nicht immer berücksichtigt. Zum Beispiel ist es für den Benutzer nicht oder nur schwer erkennbar, ob das Verschieben und Fallenlassen eines Dokumentes über der Toolbox eines „entfernten Desktops“ zum Start einer Anwendung oder lediglich zur Positionierung des Dokumentes auf den entfernten Desktop führt. Die Funktionalität der Toolbox „Benutzer“ hingegen, ist nur auf das Verteilen von Dokumenten begrenzt: Für den Benutzer ist es „vorhersehbar“ (Erwartungskonformität), dass das Dokument nach der Interaktion mit der Toolbox auch tatsächlich bei dem anderen angemeldeten Benutzer erscheint.

Die Beeinflussungsmöglichkeiten für den angemeldeten Benutzer, in Bezug auf die Erreichbarkeit, die Verteilung und den Zugriff, sind nur begrenzt oder nicht vorhanden:

- Erreichbarkeit: Die Verbindungen zu den Services wird aktiv von dem Benutzer vorgenommen. Der Verbindungsaufbau zu anderen Benutzern ist passiv: Die beteiligten Benutzer werden nicht gefragt, ob sie miteinander verbunden werden möchten - hier besteht keine Steuerungsmöglichkeit der Erreichbarkeit.
- Verteilung: Das Verteilen von Dokumenten ist aktiv. Der Empfänger hat keine Möglichkeit, zugesandte Dokumente abzulehnen. Auch hier fehlt es an Steuerungsmöglichkeiten der Verteilung seitens des Empfängers
- Zugriffs: Die Dokumente werden immer als Kopie versendet. Der Sender verliert jede Steuerungsmöglichkeit des Zugriffs (und auch gesonderte Rechte) bei dieser Kopie.

Der Prototyp wurde bisher nur mit wenigen gleichzeitig angemeldeten Benutzern getestet. Wie verhält sich aber das System, wenn die Anzahl der Teilnehmer und auch der zur Verfügung gestellten „entfernten Desktops“ erheblich erhöht wird? Ist das CSCW-System für den einzelnen Benutzer dann noch handhabbar?

## 5.2. Ausblick

Die mit diesem Prototyp vorgestellte „Multi-Desktop-Steuerung“ ist die erste lauffähige Version nach einem Prototyping-Zyklus. Für die nachfolgende Version müsste, abgesehen von den einfließenden Ergebnissen einer software-ergonomischen Bewertung durch Test-Benutzer und auch Usability-Tests, die Steuerung entfernter Desktops ausgebaut werden. Derzeit wird diese zum größten Teil nur simuliert. Besonders interessant, mag hier die Verschiebung von ganzen Anwendungen über Desktop-Grenzen hinweg sein: Der angemeldete Benutzer stellt eine lokal, auf seinem Laptop, geöffnete Anwendung anderen Benutzern, beispielsweise über die Powerwall, zur Verfügung. Obwohl der CSCW-Bereich ein noch recht junger Informatikzweig ist, sind schon einige Gesamtsysteme für CSCW-Konferenzräume entwickelt worden. Ich denke, dass in naher Zukunft CSCW-Konferenzräume zur Standardausstattung von Unternehmen gehören werden. Für die Motivation CSCW einzusetzen, spielen vor allem wirtschaftlichen Interessen eine große Rolle (beschrieben in Unterkapitel 2.1.2 S.19).

# Literaturverzeichnis

- [Bartnik 2006] BARTNIK, Roman: *Diplomarbeit Weiterentwicklung einer Technologiebasis für interaktive Gruppenarbeitsräume*. HAW Hamburg, 2006
- [Collomb und Hascoet 2004] COLLOMB, Maxime ; HASCOET, Mountaz: *Speed and accuracy in throwing models*. Le Laboratoire d' Informatique de Robotique et de Microélectronique de Montpellier, 2004. – URL <http://www.lirmm.fr/~collomb/publi/hci2004collomb-hascoet.pdf>
- [Grudin u. a. 1995] GRUDIN, Jonathan ; BAECKER, Ronald M. ; BUXTON, William: *Readings in Human-Computer Interaction: Toward the Year 2000*. Morgan Kaufmann Publishers Inc,US, 1995. – ISBN 978-1558602465
- [Hermann u. a. 1994] HERMANN, T. ; HARTMANN, A. ; ROHDE, M. ; WULF, V.: *Menschengerechte Groupware. Software-ergonomische Gestaltung und partizipative Umsetzung*. Teubner, 1994. – ISBN 978-3519026839
- [Hewett u. a. 2007] HEWETT ; BAECKER ; CARD ; CAREY ; GASEN ; MANTEI ; PERLMAN ; STRONG ; VERPLANK: *AACM SIGCHI Curricula for Human-Computer Interaction*. (2007). – URL [http://sigchi.org/cdg/cdg2.html#2\\_1](http://sigchi.org/cdg/cdg2.html#2_1)
- [Hinckley ] HINCKLEY, Ken: *Bumping Objects Together as a Semantically Rich Way of Forming Connections between Ubiquitous Devices*. . – URL <http://research.microsoft.com/users/kenh/papers/Ubicomp03Abstract.pdf>
- [IPIS 2007] IPIS, Fraunhofer: *Projekte*. (2007). – URL <http://www.ipsi.fraunhofer.de/ambiente/projekte/projekte/index.html>
- [Johansen u. a. 1991] JOHANSEN, Robert ; MARTIN, Alixia ; SIBBET, David: *Leading Business Teams: How Teams Can Use Technology and Group Process Tools to Enhance Performance*. Addison Wesley, 1991. – ISBN 978-0201528299
- [Lahlou und Jegou 2004] LAHLOU, Saadi ; JEGOU, François: *European Disappearing Computer Privacy Design Guidelines, Version 1.1*. (2004). – URL <http://www.rufae.net/privacy.html>
- [Mund 2006] MUND, Horst: *Diplomarbeit Berechtigungsstrukturen in kollaborativen Umgebungen*. HAW Hamburg, 2006

- [Neuman 2006] NEUMAN, Carola: *Diplomarbeit Effizienzsteigerung konzeptioneller Diskussions- und Entscheidungsprozesse*. HAW Hamburg, 2006
- [Norman 2002] NORMAN, Donald A.: *The Design of Everyday Things*. B And T, 2002. – ISBN 978-0465067107
- [Schlichter und Borghoff 1998] SCHLICHTER, Johann H. ; BORGHOFF, Uwe M.: *Rechnergestützte Gruppenarbeit. Eine Einführung in Verteilte Anwendungen*. Springer, 1998. – ISBN 978-3540628736
- [Tandler 2002] TANDLER, Peter: *The BEACH Application Model and Software Framework for Synchronous Collaboration in Ubiquitous Computing Environments*. Fachbereich Informatik der Technischen Universität Darmstadt, 2002
- [Tandler 2004] TANDLER, Peter: *Synchronous Collaboration in Ubiquitous Computing Environments*. Dissertation vom Fachbereich Informatik der Technischen Universität Darmstadt, 2004
- [Weiser 1991] WEISER, Marc: *The computer for the 21st century*. Scientific American, 1991
- [Wilson 1991] WILSON, Paul: *Computer Supported Cooperative Work: An Introduction*. Kluwer Academic Publishers, 1991. – ISBN 978-0792314462

# A. Tabellen der Anwendungsfälle

<b>Anwendungsfall 1:</b>	<b>Systemanmeldung</b>
<b>Zielbeschreibung:</b>	Ein Benutzer meldet sich bei unterschiedlichen Diensten/Servern an und steht anderen Benutzern zwecks virtueller Interaktionen zur Verfügung.
<b>Vorbedingung:</b>	Benutzerkonto existiert; Server/Services sind gestartet.
<b>Endbedingungen:</b>	
Erfolgreich:	Benutzerkonto ist für andere Benutzer sichtbar; Benutzer sieht erfolgreichen Verbindungsaufbau
Nicht erfolgreich:	Benutzerkonto ist für andere Benutzer nicht sichtbar
<b>Hauptakteure:</b>	Benutzer, Server/Service
<b>Beschreibung (Schritt / Aktion):</b>	<ol style="list-style-type: none"><li>1. Service-Auswahl</li><li>2. Eingabe Benutzername</li><li>3. Eingabe Passwort</li><li>4. Starten des/der Services</li></ol>
<b>Variation (ab Schritt / Aktion):</b>	
<b>Erweiterung (ab Schritt / Aktion):</b>	

Tabelle A.1.: Anwendungsfall 1: Systemanmeldung

<b>Anwendungsfall 2:</b>	<b>Bereitstellung von Artefakten</b>
<b>Zielbeschreibung:</b>	Der Benutzer stellt in einem speziellen Kontainer Dokumente etc. für die Sitzung bereit. Er kann die Dokument von seiner Desktop-Arbeitsfläche oder aus einem Datei-Browser via Drag-And-Drop in diesen Kontainer ablegen. Der Kontainer ist nur für ihn zugreifbar.
<b>Vorbedingung:</b>	Es besteht ein Ziel-Pfad - einzustellen bei der Konfiguration - der zu den realen Kontainer (Verzeichnis) zeigt.
<b>Endbedingungen:</b>	
Erfolgreich:	Dokument ist in dem Kontainer zu sehen.
Nicht erfolgreich:	Kontainer enthält nicht gewähltes Dokument.
<b>Hauptakteure:</b>	Benutzer
<b>Beschreibung (Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl eines Dokumentes</li> <li>2. Initiierung einer Drag-And-Drop-Aktion</li> <li>3. Drop-Kontainer wählen</li> <li>4. Beenden der Drag-Drop-Aktion</li> </ol>
<b>Variation (ab Schritt / Aktion):</b>	
<b>Erweiterung (ab Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl mehrerer Dokumente</li> <li>2. wie oben</li> </ol>

Tabelle A.2.: Anwendungsfall 2: Bereitstellung

Anwendungsfall 3:	Verteilung von Artefakten
<b>Zielbeschreibung:</b>	Der Benutzer versendet Dokumente an andere angemeldete Benutzer (aktives Senden). Die Versendung kann sowohl vom eigenen Arbeitsplatz erfolgen als auch vom Server/Service aus.
<b>Vorbedingung:</b>	Andere Benutzer sind angemeldet.
<b>Endbedingungen:</b>	Erfolgreich: Dokument taucht im Projekt-Kontainer des Empfängers auf. Nicht erfolgreich: Das Dokument wurde nicht übertragen und befindet sich nicht im Projekt-Kontainer des Empfängers.
<b>Hauptakteure:</b>	Benutzer, Server/Service
<b>Beschreibung (Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl eines Dokumentes</li> <li>2. Initiierung einer Drag-And-Drop-Aktion</li> <li>3. Drop-Kontainer eines Benutzers wählen</li> <li>4. Beenden der Drag-Drop-Aktion</li> </ol>
<b>Variation (ab Schritt / Aktion):</b>	
<b>Erweiterung (ab Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl mehrerer Dokumente</li> <li>2. wie oben</li> </ol>

Tabelle A.3.: Anwendungsfall 3: Verteilung

<b>Anwendungsfall 4: Initiierung eines Bearbeitungsprozesses</b>	
<b>Zielbeschreibung:</b>	Bearbeitung respektive Anzeigen von Dokumente mittels entfernter Geräte. Einzelne Dokumente werden vom Benutzer an Server/Service gesendet, die unterschiedliche sowohl Ausgabe-, Eingabegeräte steuern als auch zu den Dokumententyp passende Anwendungen zur Verfügung stellen.
<b>Vorbedingung:</b>	Server/Service ist gestartet, Anwendung zum Dokumententyp ist vorhanden
<b>Endbedingungen:</b>	
Erfolgreich:	Passende Anwendung beim Server/Service mit Dokument gestartet (sichtbar).
Nicht erfolgreich:	Keine Anwendung beim Server/Service gestartet
<b>Hauptakteure:</b>	Benutzer, Server/Service
<b>Beschreibung (Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl eines Dokumentes</li> <li>2. Initiierung einer Drag-And-Drop-Aktion</li> <li>3. Drop-Kontainer eines Servers/Services wählen</li> <li>4. Beenden der Drag-Drop-Aktion</li> </ol>
<b>Variation (ab Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl eines Servers/Services</li> <li>2. Auswahl eines Dokumentes</li> <li>3. Initiierung einer Drag-And-Drop-Aktion</li> <li>4. Drop-Kontainer des Servers/Services wählen</li> <li>5. Beenden der Drag-Drop-Aktion</li> </ol>
<b>Erweiterung (ab Schritt / Aktion):</b>	

Tabelle A.4.: Anwendungsfall 4: Initiierung eines Bearbeitungsprozesses

<b>Anwendungsfall 5:</b>	<b>Administration des Servers</b>
<b>Zielbeschreibung:</b>	Hinzufügen und Konfigurieren von Anwendungen, die innerhalb des Server/Services zur Verfügung stehen sollen. Verbindungsspezifische und service-spezifische Einstellungen
<b>Vorbedingung:</b>	Anwendungen müssen vorhanden sein
<b>Endbedingungen:</b>	
Erfolgreich:	Einstellungen wurden übernommen.
Nicht erfolgreich:	Einstellungen wurden nicht übernommen
<b>Hauptakteure:</b>	Administrator
<b>Beschreibung (Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl einer Einstellungsmöglichkeit</li> <li>2. Verändern</li> <li>3. Speichern der Einstellungen</li> </ol>
<b>Variation (ab Schritt / Aktion):</b>	
<b>Erweiterung (ab Schritt / Aktion):</b>	

Tabelle A.5.: Anwendungsfall 5: Administration des Servers

<b>Anwendungsfall 6:</b>	<b>Konfiguration des Clients</b>
<b>Zielbeschreibung:</b>	Verbindungsspezifische und persönliche Einstellungen (Bild)
<b>Vorbedingung:</b>	
<b>Hauptakteure:</b>	Benutzer
<b>Endbedingungen:</b>	
Erfolgreich:	Einstellungen wurden übernommen
Nicht erfolgreich:	Einstellungen wurden nicht übernommen
<b>Beschreibung (Schritt / Aktion):</b>	<ol style="list-style-type: none"> <li>1. Auswahl einer Einstellungsmöglichkeit</li> <li>2. Verändern</li> <li>3. Speichern der Einstellungen</li> </ol>
<b>Variation (ab Schritt / Aktion):</b>	
<b>Erweiterung (ab Schritt / Aktion):</b>	

Tabelle A.6.: Anwendungsfall 6: Konfiguration des Clients

# B. Prototyp-Abbildungen

## B.1. Abbildungen: ProjectRoomClientApp



Abbildung B.1.: ProjectRoomClientApp: Gestartete Anwendung

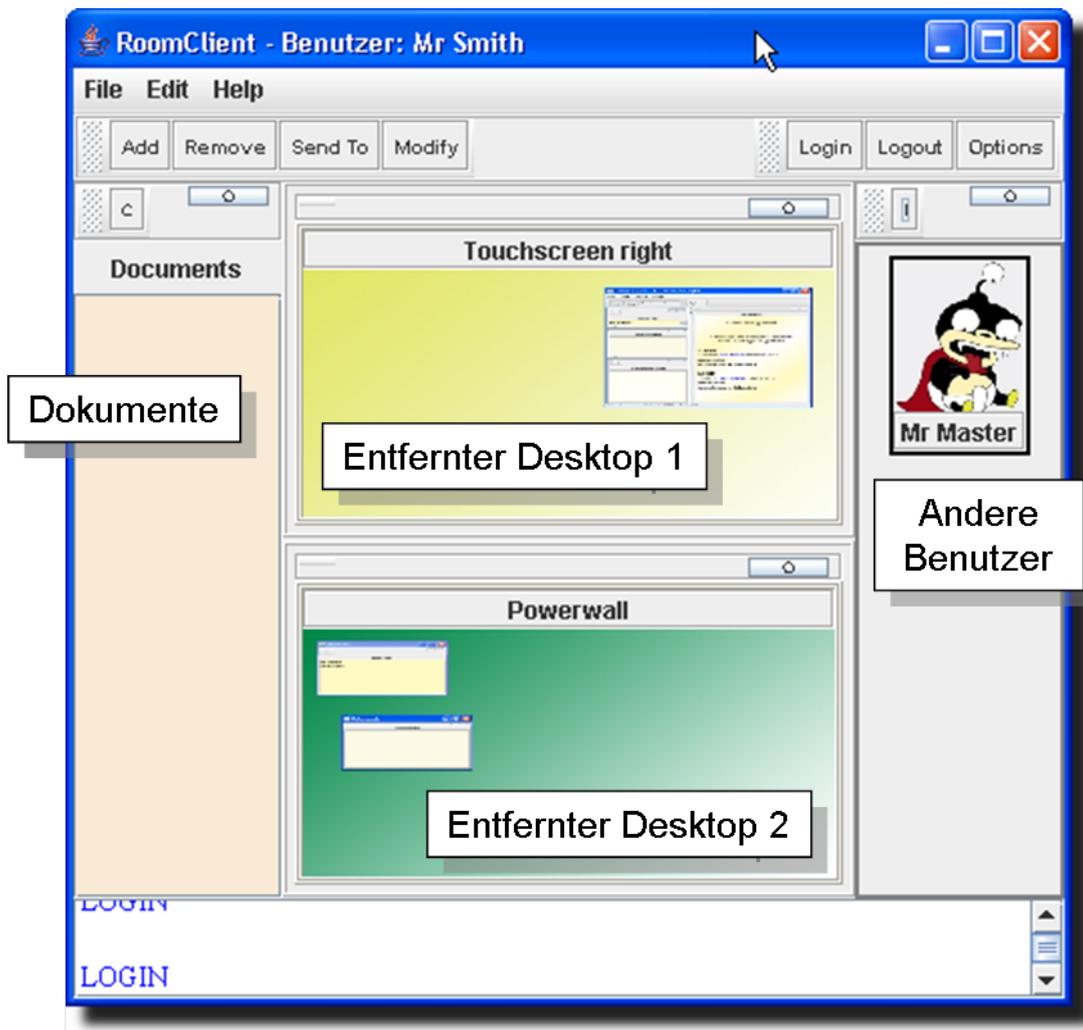


Abbildung B.2.: ProjectRoomClientApp - Der Benutzer hat sich angemeldet

### B.1.1. Abbildungen: Toolbox

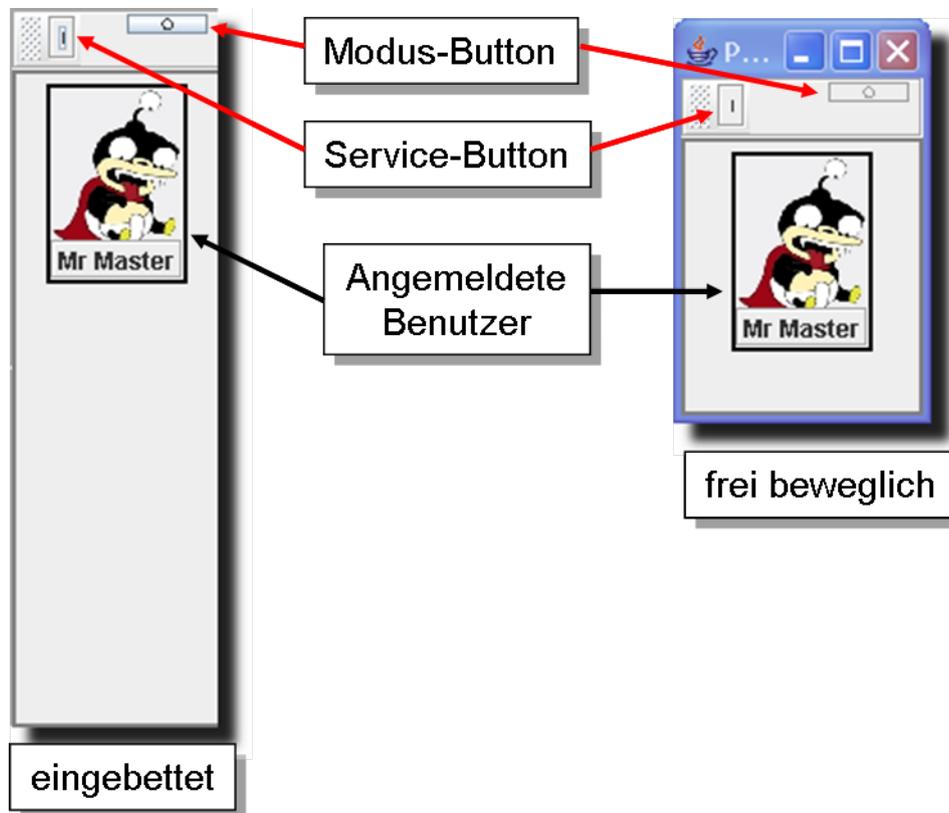


Abbildung B.3.: Toolbox - Freier und eingebetteter Modus

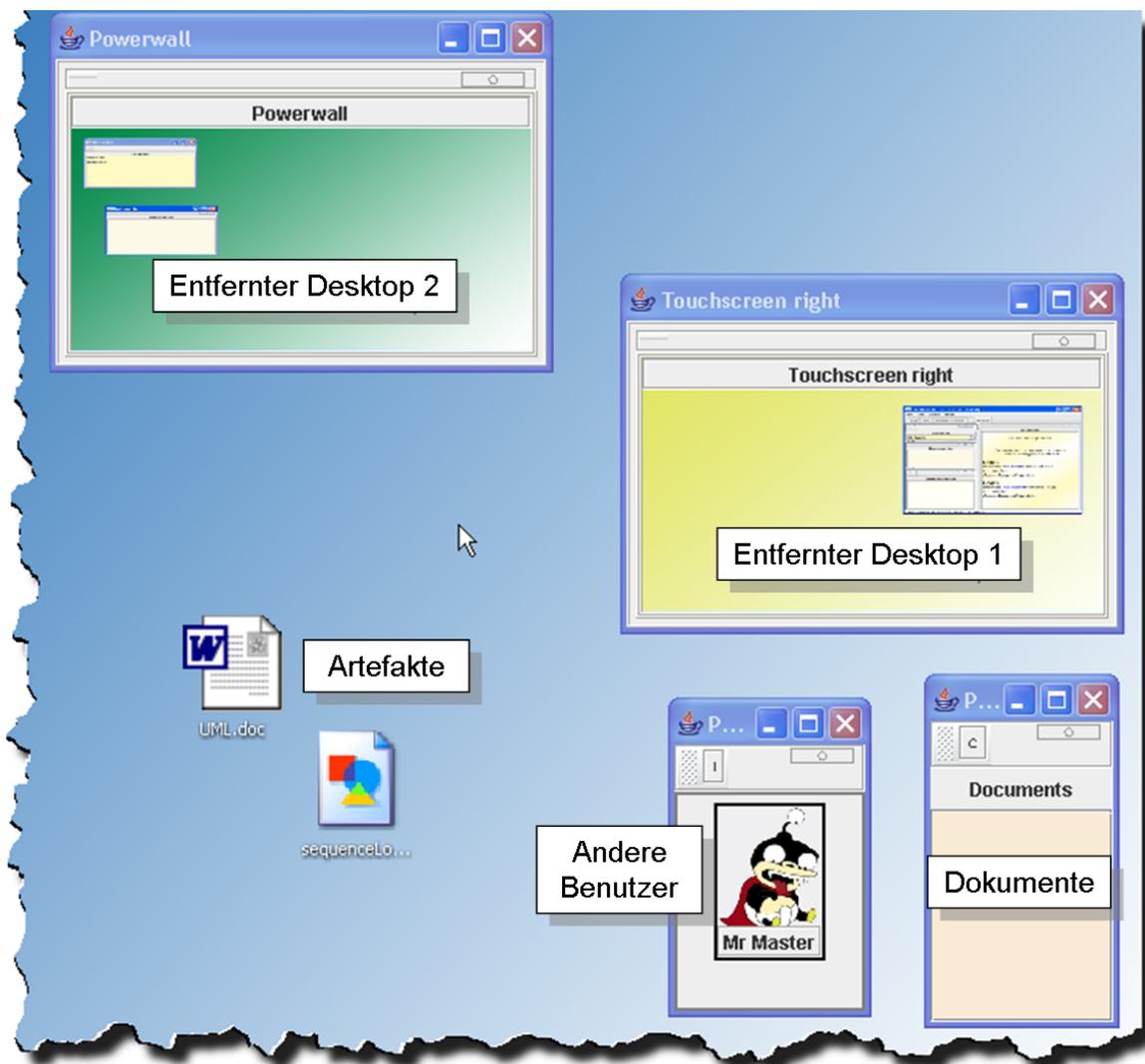


Abbildung B.4.: ProjectRoomClientApp - Die Toolboxen liegen (frei verschiebbar) auf dem lokalen Desktop

## B.2. Abbildungen: ProjectRoomServer

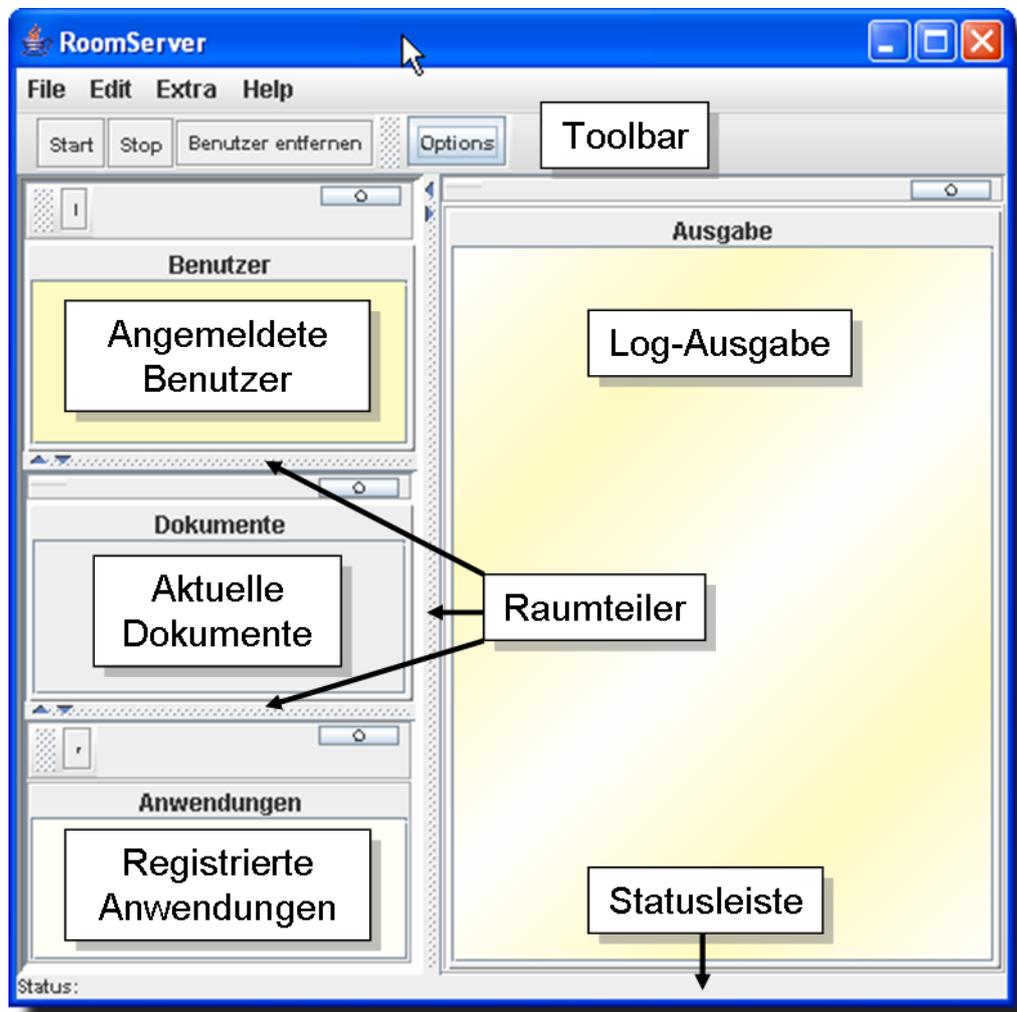


Abbildung B.5.: ProjectRoomServer - Gestartete Anwendung



Abbildung B.6.: ProjectRoomServer - Gestartete Anwendung und Service

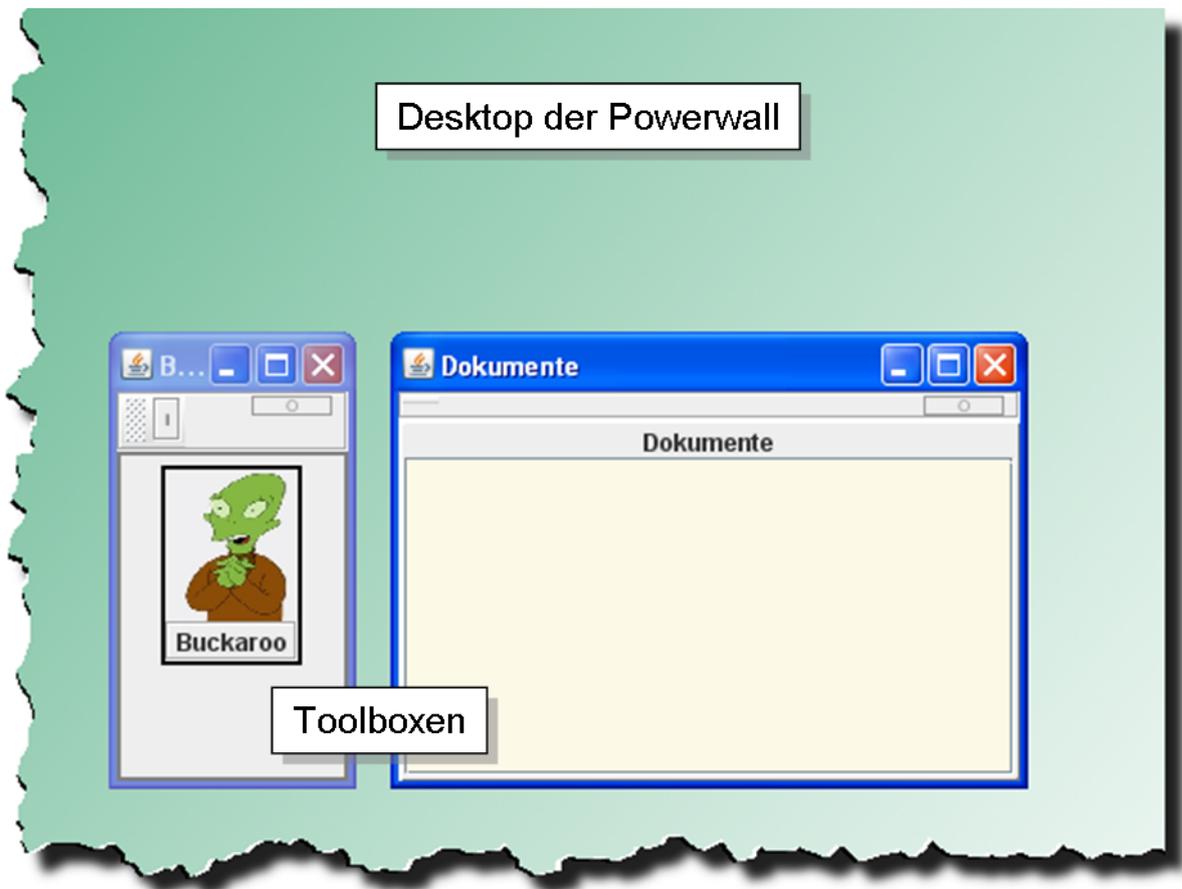


Abbildung B.7.: ProjectRoomServer - Verteilte Toolboxen auf dem Server-Desktop

### B.3. Abbildungen: Interaktionen

#### B.3.1. Benutzer-Anmeldung

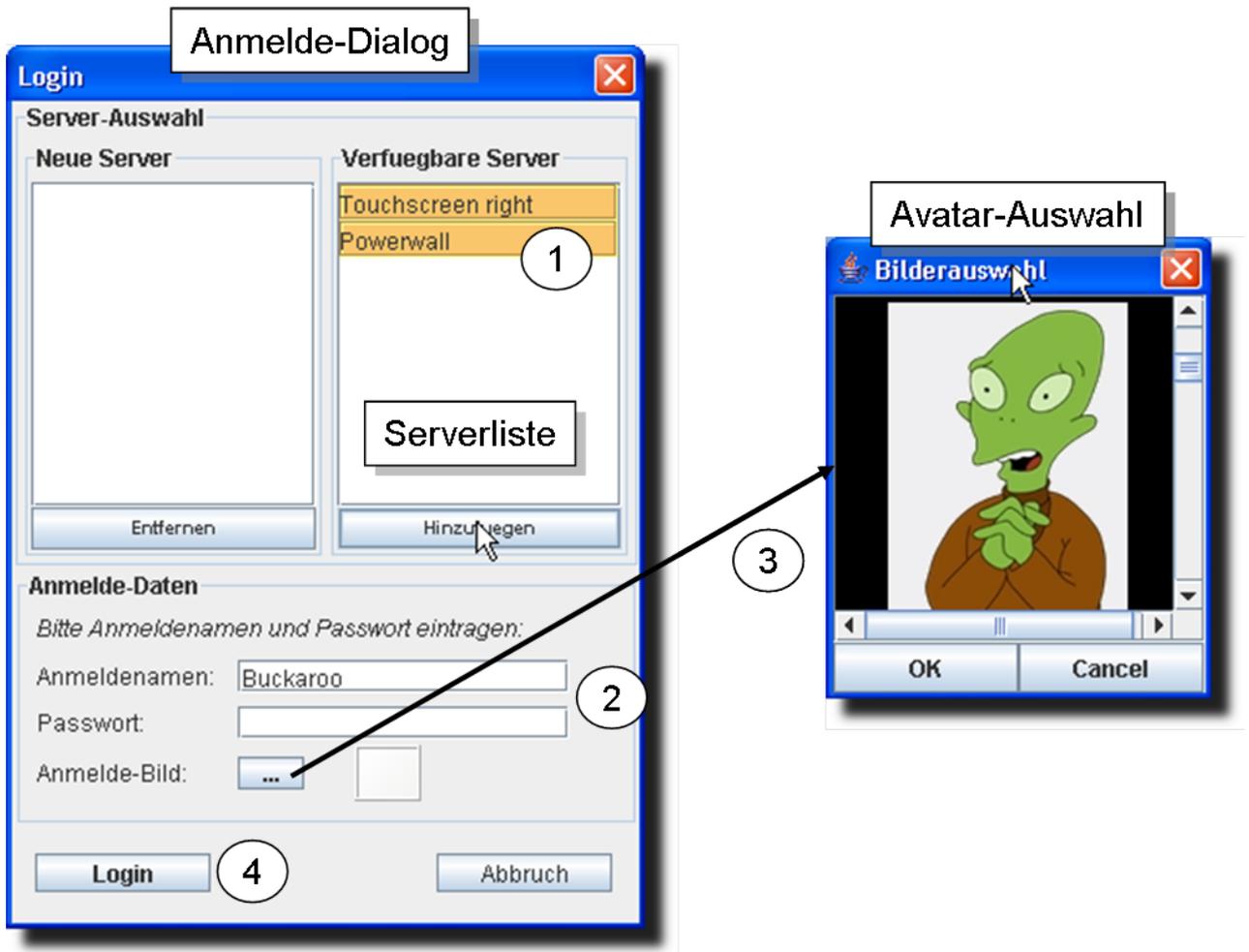


Abbildung B.8.: Benutzer-Anmeldung - Dialog zur Anmeldung

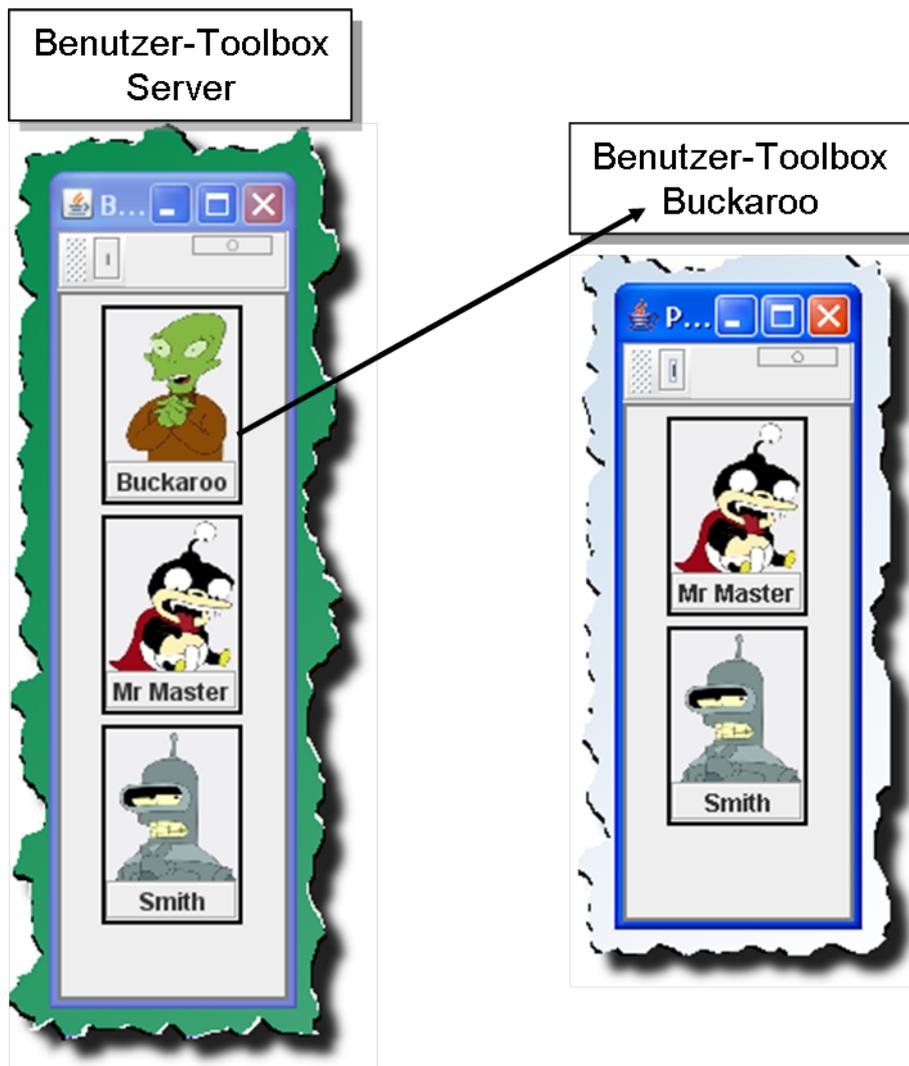


Abbildung B.9.: Benutzer-Anmeldung - Inhalt der Benutzer-Toolboxen

### B.3.2. Dokument transferieren und Initiierung eines Bearbeitungsprozesses



Abbildung B.10.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop der Powerwall vor der Ausführung

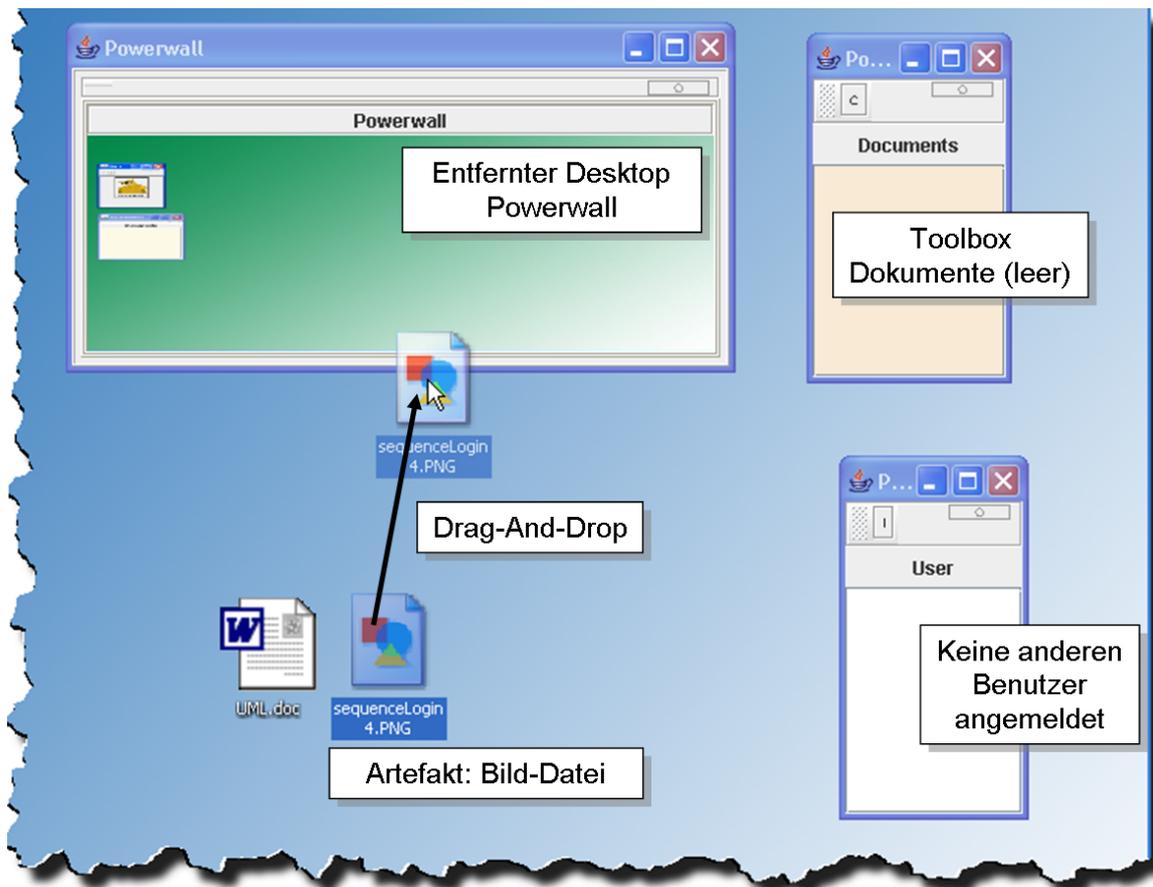


Abbildung B.11.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers und der Verschiebevorgang

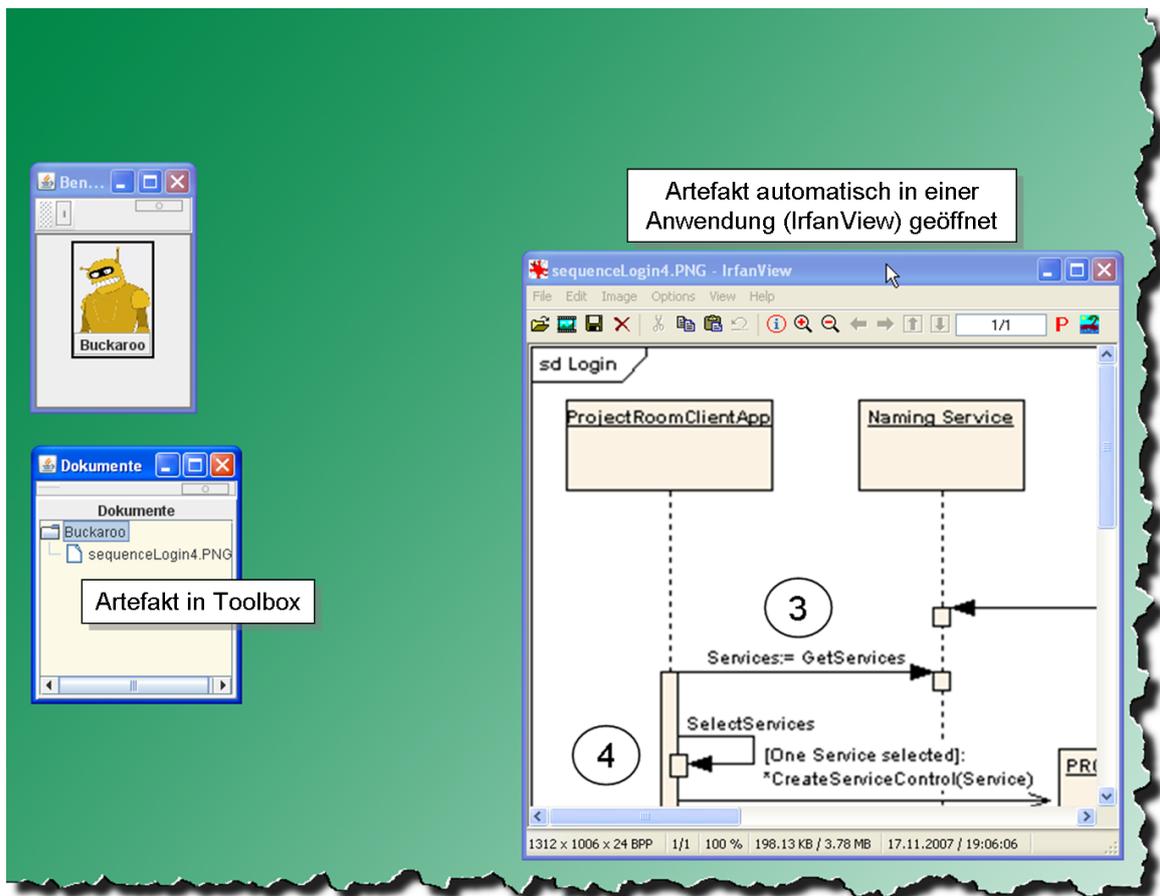


Abbildung B.12.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses  
- Anzeige des Bildes mit einer Anwendung (IrfanView), aktualisierte  
Dokumente-Toolbox

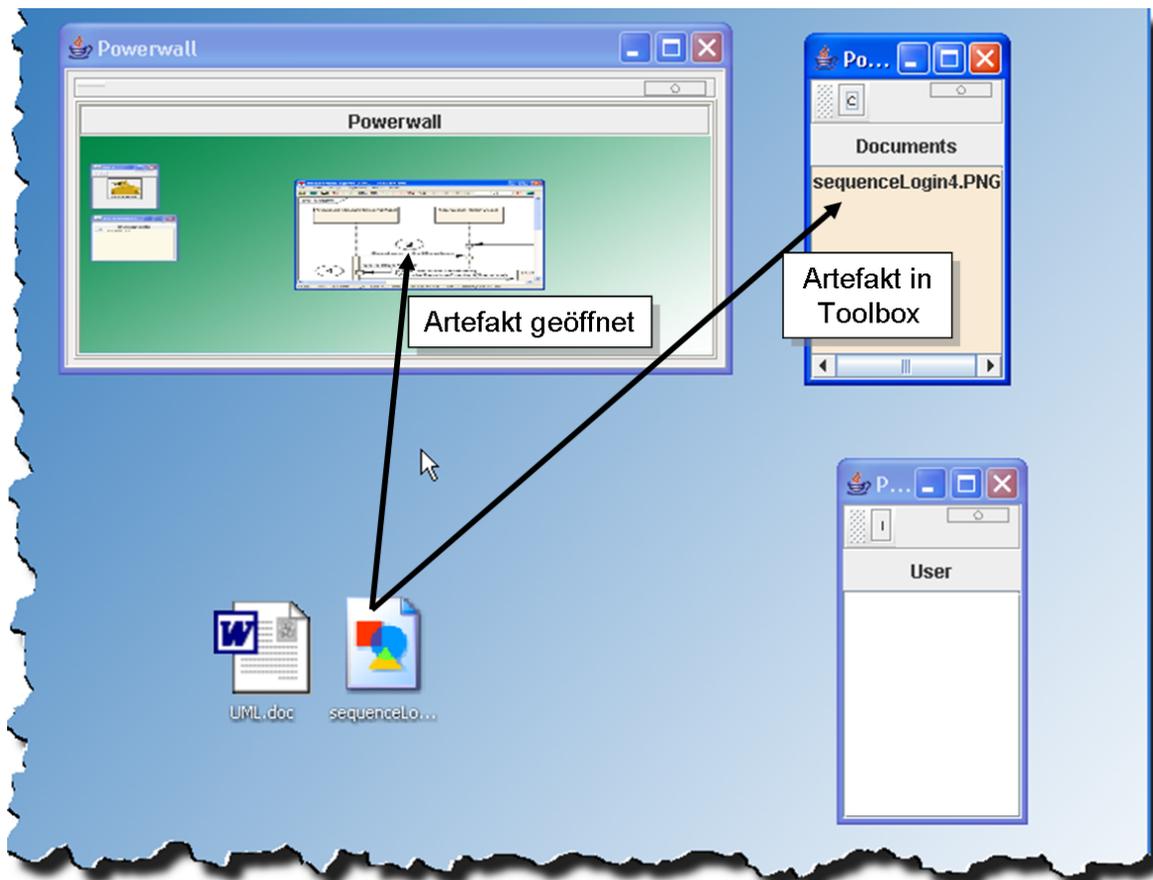


Abbildung B.13.: Dokument transferieren und Initiierung eines Bearbeitungsprozesses - Desktop des Benutzers nach der Aktion

### B.3.3. Verteilen von Dokumenten

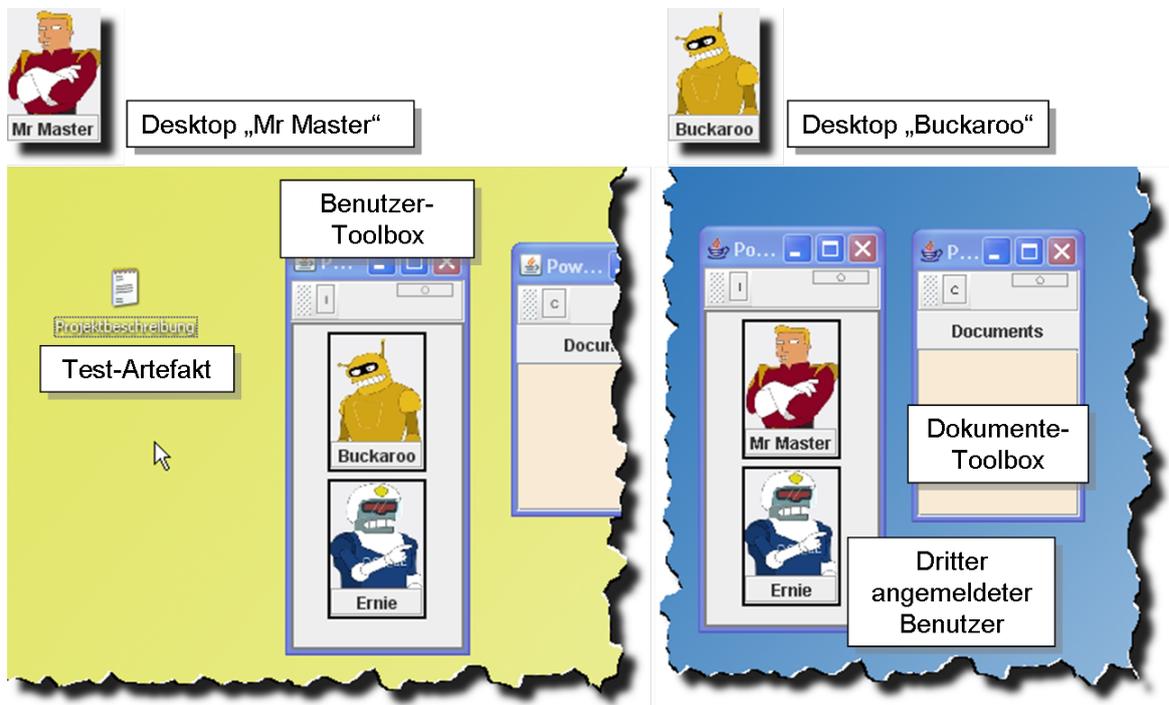


Abbildung B.14.: Verteilen von Dokumenten - Die Desktops der beiden Benutzer vor der Verteilung

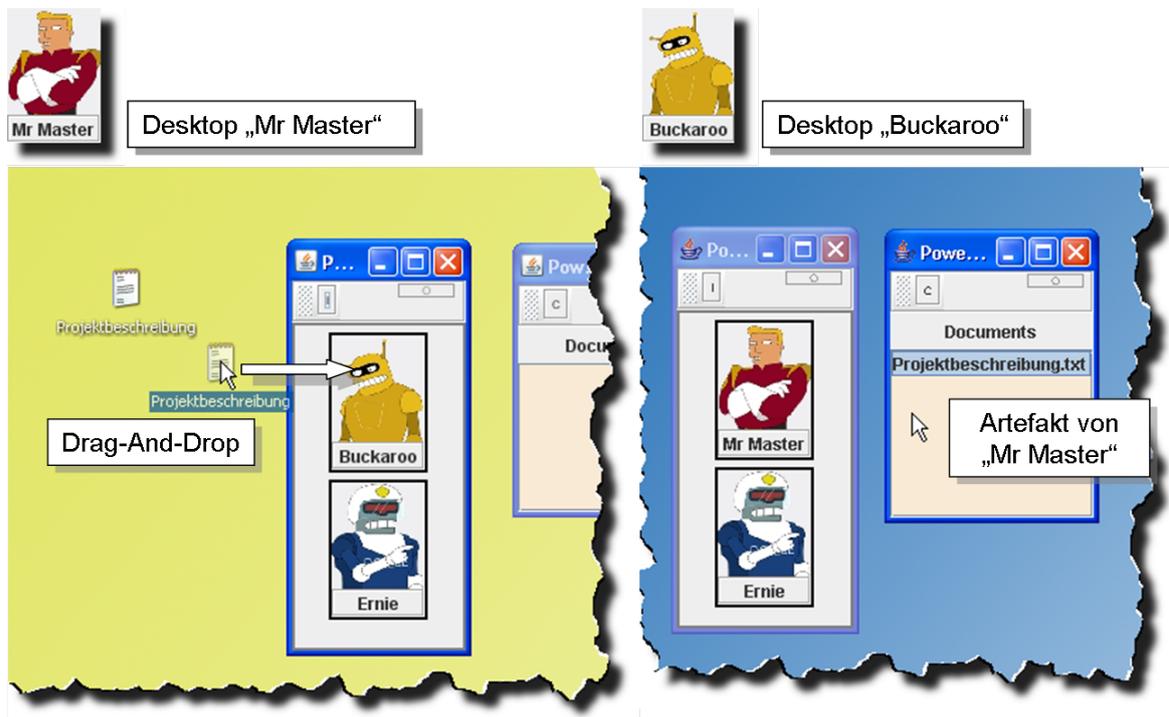


Abbildung B.15.: Verteilen von Dokumenten - Das Dokument taucht in der Dokumenten-Toolbox von „Buckaroo“ auf.

## B.3.4. Zusatz - Konfiguration/Administration der Anwendungen

### B.3.4.1. Konfiguration von ProjectRoomClientApp

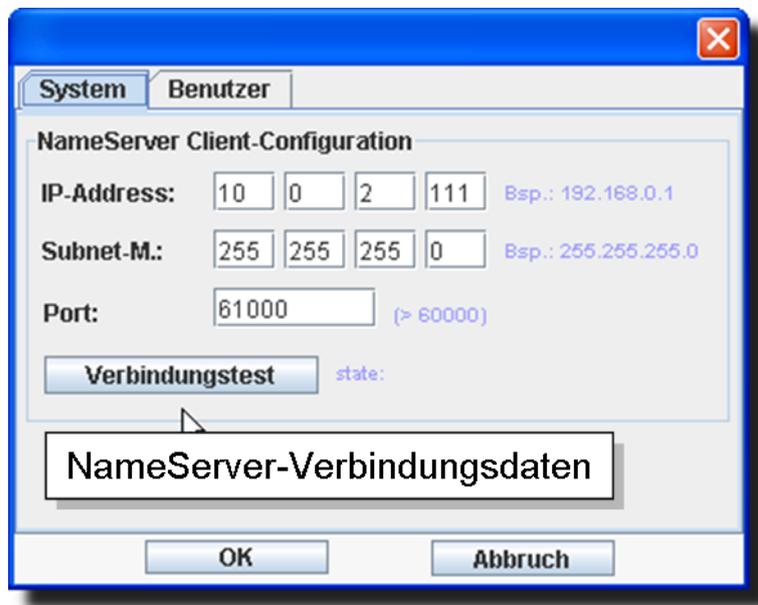


Abbildung B.16.: Konfiguration von ProjectRoomClientApp - NameServer-Verbindungseinstellungen



Abbildung B.17.: Konfiguration von ProjectRoomClientApp - Benutzereinstellungen

**B.3.4.2. Administration von ProjectRoomServer**

Abbildung B.18.: Administration von ProjectRoomServer - NameServer-Verbindungseinstellungen



Abbildung B.19.: Administration von ProjectRoomServer - Verwalten der registrierten Anwendungen

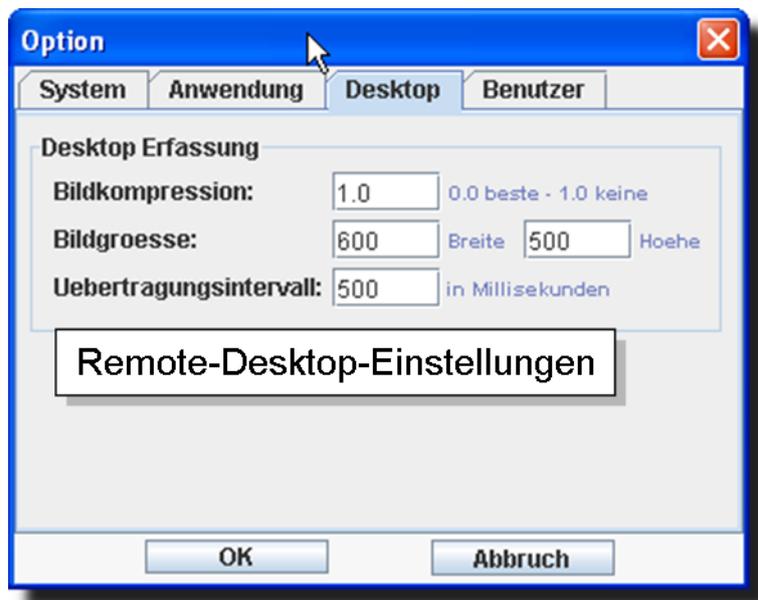


Abbildung B.20.: Administration von ProjectRoomServer - Remote-Desktop-Einstellungen

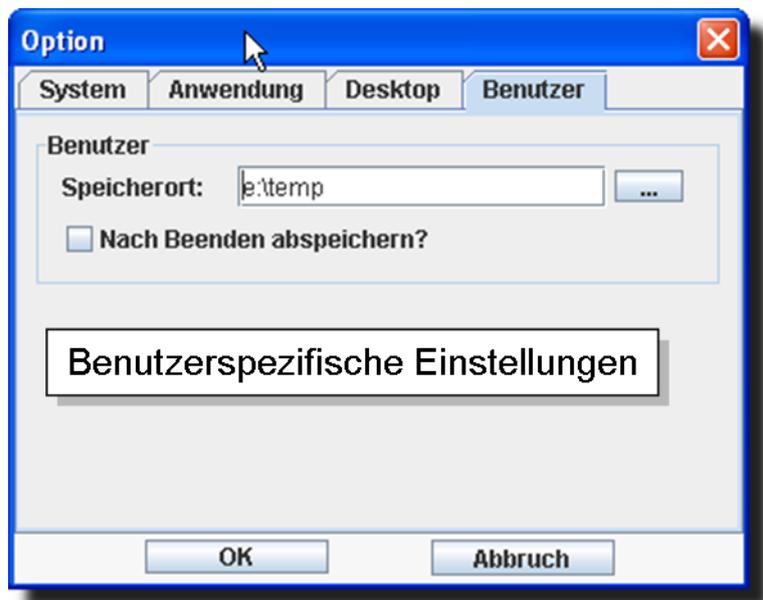


Abbildung B.21.: Administration von ProjectRoomServer - Benutzereinstellungen

The image shows a Windows-style dialog box titled "Registrierung neuer Anwendungen". The dialog is divided into several sections:

- Daten:** Contains input fields for "Name:", "Beschreibung:", "Typ:", "Display-Conte...", and "Max. Instanzen" (set to 1). There are also fields for "Programm:" (with a browse button "...") and "Parameter:".
- Sicherheit:** A large empty rectangular area for security settings, with "add" and "remove" buttons below it.
- Datei-Arten:** A section containing a box with the text "Zugehörige Datei-Erweiterungen" and "add" and "remove" buttons below it.
- Aktionen:** A section at the bottom with "OK" and "Abbruch" buttons.

Abbildung B.22.: Administration von ProjectRoomServer - Registrierung neuer Anwendungen

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 28. November 2007

Ort, Datum

Unterschrift