



Universität Regensburg

Philosophische Fakultät III
Sprach-, Literatur- und Kulturwissenschaften
Institut für Information und Medien, Sprache und Kultur (I:IMSK)
Lehrstuhl für Informationswissenschaft

Masterarbeit im Fach Informationswissenschaft

BRAINSTORMING UND MIND-MAPPING IM MULTI-DEVICE-KONTEXT

Konzeption und prototypische Implementierung
für Multi-Touch-Tabletop und Smartphone

vorgelegt von: Sarah Will, B.A.

Erstgutachter: Prof. Dr. Christian Wolff

Zweitgutachter: Prof. Dr. Rainer Hammwöhner

Laufendes Semester: Wintersemester 2011/12

vorgelegt am: 11. Januar 2012

Zusammenfassung

Die vorliegende Arbeit beschreibt die Konzeption und prototypische Implementierung einer Anwendung zur elektronischen Unterstützung von Brainstorming- und Mind-Mapping-Sitzungen an einem multitouchfähigen Tabletop mit Smartphones. Während der Tabletop durch seine große, horizontal ausgerichtete Oberfläche die kollaborative Erstellung und Strukturierung von Ideen in Gruppen unterstützt, werden Smartphones einen zusätzlichen Eingabekanal zur Verfügung stellen sowie Individualarbeit fördern. Somit wird zunächst einerseits die Motivation für die Verwirklichung einer Anwendung zum Brainstorming und Mind-Mapping am Tabletop und andererseits das Potential einer Ergänzung eines solchen Systems durch zusätzliche private Geräte im Sinne der Multi-Device-Interaktion erläutert werden. Um eine geeignete theoretische Basis zu schaffen, wird darüber hinaus ein detaillierter Überblick über die Entwicklung der Tabletop-technologie in den letzten Jahren und den dabei zentralen Forschungs- und Problemfeldern im Kontext der Entwicklung von Benutzerschnittstellen für Tabletopsysteme gegeben. Zudem werden schließlich auch im Rahmen der Arbeit relevante Aspekte der Multi-Device-Interaktion skizziert. Diesem theoretisch ausgerichteten Teil der Arbeit folgt schließlich eine Beschreibung der Entwicklung des Anwendungskonzepts und der dabei formulierten Anforderungen und Designzielen, welche in Form eines ersten, papierbasierten Prototypen visualisiert werden. Aufbauend auf diesen konzeptuellen Überlegungen wird schließlich die konkrete technische Umsetzung der Anwendung *Multi/Touch/Device Mind-Mapper* in Form eines *High-Level*-Prototypen auf Basis des *MT4j*-Frameworks (Tabletop) und des *Android*-Betriebssystems (Smartphone) beleuchtet. Eine Diskussion dieses finalen Prototypen und erster praktischer Erfahrungen sowie ein Ausblick auf Erweiterungsmöglichkeiten des Systems und über den Rahmen der Arbeit hinaus gehende Fragestellungen schließen die Arbeit ab. Schlussendlich kann gezeigt werden, dass die im Rahmen des Anwendungskonzept definierten Anforderungen mit Hilfe der verwendeten Frameworks bis auf wenige Ausnahmen erfolgreich umgesetzt werden konnten. Darüber hinaus kann der Anwendung durch die aus dem Testbetrieb gewonnenen Erkenntnisse eine grundsätzliche Praktikabilität attestiert, ebenso konnten einige Ansatzpunkte für weitere Verbesserungen und Tests aufgedeckt werden.

Abstract

This work describes the conceptual design and implementation of a prototypic application for the electronic support of brainstorming and mind mapping sessions with a multi-touch tabletop and smartphones. While the large, horizontally aligned, flat surface of the tabletop supports the collaborative creation and structuring of ideas in groups, smartphones provide an additional interaction channel and assist individual work. First of all, this thesis sets out to illustrate the motivation behind the realization of an application for brainstorming and mind mapping on a tabletop as well as to point out the potential of complementing a system of that kind with additional private devices within the context of multi-device interaction. In order to establish an adequate theoretical basis, a detailed overview over the development of tabletop technology in the last years as well as the central research areas and issues with regards to the development of user interfaces for tabletops will be given. Relevant aspects of multi-device interaction will be outlined as well. Following this theoretically grounded part is the specification of the application concept including specific requirements and design goals, which will be incorporated into a first paper prototype. Building upon this conceptual work, the technical implementation of the *Multi/Touch/Device MindMapper* application, which is based on the *MT4j* framework (tabletop) and the *Android* operating system (smartphones), will be illustrated. A discussion of this final prototype and first findings from the trial period as well as questions that exceed the scope of this work complete the thesis. Ultimately, this work shows the successful implementation of most of the previously defined requirements with the aid of the used frameworks. Furthermore, the insights gained during the initial test phase attest to a certain practicability of the application, whereas several key issues relevant for future revisions and testing could be identified as well.

Inhaltsverzeichnis

Abbildungsverzeichnis.....	v
Abkürzungsverzeichnis.....	x
1 Einleitung.....	I
2 Brainstorming und Mind-Mapping im Kontext der CSCW.....	4
2.1 Kreativitätstechniken und Computer Supported Cooperative Work.....	4
2.2 Theoretische Grundlagen zum Brainstorming und Mind-Mapping.....	7
2.3 Gruppenunterstützende Systeme zur Ideengenerierung und -strukturierung	13
2.4 Motivation für das Brainstorming und Mind-Mapping am Tabletop.....	16
3 Interaktive Tabletops und Multi-Touch-Interaktion.....	19
3.1 Entwicklung der Tabletoptechnologie.....	19
3.2 Problem- und Forschungsfelder.....	32
3.2.1 Tabletopdesign.....	33
3.2.2 Aspekte der Touch-Interaktion.....	37
3.2.3 Benutzerschnittstellengestaltung für Tabletops.....	44
3.2.4 Kollaborative Nutzung von Tabletops.....	56
3.3 Brainstorming und Mind-Mapping am Tabletop.....	64
4 Multi-Device-Interaktion mit Tabletop und Smartphones.....	71
4.1 Single Display Groupware, Multi-Display-Umgebungen und Multi-Device-Interaktion.....	71
4.1.1 Definitionen und Abgrenzung.....	71
4.1.2 Multi-Device-Anwendungen in kollaborativen Szenarien.....	73
4.2 Motivation für die Kombination von Tabletop und Smartphones.....	77

5 Anwendungskonzept.....	82
5.1 Anforderungsbeschreibung.....	83
5.1.1 Aufgabenanalyse.....	83
5.1.2 Funktionale Anforderungen.....	84
5.2 Implementierungskonzept.....	85
5.3 Designziele und Lösungsansätze.....	88
5.3.1 Screenkonzept.....	88
5.3.2 Gestenalphabet.....	89
5.3.3 Benutzerschnittstellengestaltung	92
5.4 Erweiterungsmöglichkeiten.....	96
5.5 Anwendungsskizzen und Papierprototyp	99
5.5.1 Papierprototyp Tabletop.....	99
5.5.2 Skizzen Smartphone.....	104
6 Implementierung.....	106
6.1 Technische Grundlagen.....	106
6.2 Anwendungsstruktur.....	110
6.2.1 Architekturmodell.....	110
6.2.2 Paket- und Klassenstruktur.....	113
6.3 Konzeptänderungen und Probleme im Laufe der Implementierung.....	138
7 Diskussion.....	142
7.1 Erste Erfahrungen aus dem Testbetrieb.....	142
7.2 Ausblick.....	145
8 Fazit.....	147
Anhang.....	149
Literaturverzeichnis.....	I

Abbildungsverzeichnis

Abbildung 1: Modell des kreativen Problemlösungsprozesses nach Tassoul und Buijs (Van Boeijen & Daalhuizen 2010).....	5
Abbildung 2: (Nicht-hierarchische) Mind-Map um den Begriff „Glück“ (Buzan & Buzan 2005, S. 81).....	8
Abbildung 3: Potentielle Effekte von elektronischen Meetingsystemen (EMS) auf Prozessgewinne (<i>Process Gains</i>) und -verluste (<i>Process Losses</i>) im Gruppenkontext nach Nunamaker et al. (Nunamaker et al. 1991, S. 48).....	14
Abbildung 4: Benutzeroberfläche (Morris 2008) und Aufbau (Xerox 2011) des 1991 von Wellner entwickelten <i>DigitalDesk</i>	21
Abbildung 5: Benutzeroberfläche des <i>metaDESK</i> mit der Anwendung <i>Tangible Geospace</i> (Ullmer 2002, S. 75).....	22
Abbildung 6: Erster Prototyp des <i>InteracTable</i> 1998 (Müller-Tomfelde & Fjeld 2010, S. 8) und <i>i-LAND</i> -Roomware-Komponenten der zweiten Generation 1999 (Prante, Streitz & Tandler 2004, S. 48).....	23
Abbildung 7: Kollaborative Interaktion am <i>DiamondTouch</i> (Morris 2008) und schematischer Aufbau (Circle Twelve Inc 2008, S. 2).....	24
Abbildung 8: Schematischer Aufbau der <i>Frustrated Total Internal Reflection</i> (FTIR) Technik (Roth 2008).....	26
Abbildung 9: Schematischer Aufbau der <i>Diffused Illumination</i> (DI) Technik (Roth 2008).....	26
Abbildung 10: Produktbild (Microsoft 2008) und schematischer Aufbau des <i>Microsoft Surface</i> (Tzounopoulos 2007): 1) Bildschirm 2) Infrarotquelle 3) CPU 4) Projektor	27
Abbildung 11: Schematischer Aufbau des <i>FLATIR</i> -Prototyps – Kombination von FTIR, LC-Matrix und Infrarot-Sensoren (Hofer, Naef & Kunz 2009, S. 319).....	28
Abbildung 12: <i>Evoluce ONE</i> und <i>Evoluce TWO</i> (Evoluce 2011).....	29

Abbildung 13: <i>Microsoft Surface 2.0 / Samsung SUR40</i> (Microsoft 2011).....	30
Abbildung 14: Abweichung zwischen Zielpunkt und erfasstem Berührungspunkt bei Elementauswahl. a) Tatsächlicher Zielpunkt, b) Finger, c) Kontaktfläche, d) Erkannter Berührungspunkt (Holz & Baudisch 2011, S. 2502).....	38
Abbildung 15: Zustandsmodelle für Eingabe via Maus (3-stufig, links) und Touch-Interaktion (2-stufig, rechts) (Benko & Wigdor 2010, S. 252 f.; adaptiert aus Buxton 1990).....	41
Abbildung 16: Anteil der Übereinstimmung bei benutzergenerierten Gesten für 27 unterschiedliche Aktionen am Tabletop in Prozent (Wobbrock, Morris & Wilson 2009 S. 1088)	47
Abbildung 17: Perspektivisch unabhängige, kreisförmige Menüs in <i>TeamTag</i> (Ryall et al. 2006b, S. 58) und als Bestandteil einer <i>TabletopWidget</i> -Bibliothek (Graham et al. 2010).....	52
Abbildung 18: <i>Stacked Half-Pie</i> -Menü (o.l., Hesselmann, Flöring & Schmitt 2009, S. 173), <i>User Drawn Path</i> -Menü (o.r., Leithinger & Haller 2007, S. 122), <i>Multitouch Marker</i> -Menü (u.l., Lepinski, Grossmann & Fitzmaurice 2010b) und globales Ring-Menü (u.r., Kammer, Lammack & Grohl 2010, S. 260).....	53
Abbildung 19: <i>Tangible Interaction</i> am Tabletop am Beispiel des <i>reactTable</i> (Boldrin 2011).....	55
Abbildung 20: Aufteilung in individuell, gemeinsam und zur Ablage genutzte Arbeitsbereiche beim kollaborativen Arbeiten an einem traditionellen Tisch (Habelski 2004, S. 24)	57
Abbildung 21: Handgesten zur Schaffung von privaten Räumen zur Aus- (Wu & Balakrishnan 2003, S. 199) und Eingabe (Kim et al. 2010, S. 1096) sensibler Informationen am Tabletop.....	59
Abbildung 22: Technologien zur Unterscheidung und Identifizierung von Benutzern am Tabletop: <i>idWristband</i> (Meyer & Schmidt 2010, S. 277) und <i>TouchID</i> -Handschuhe (Marquardt et al. 2011, S. 9).....	61
Abbildung 23: <i>BeachMap</i> an der <i>DynaWall</i> (Prante, Magerkurth & Streit 2002, S. 112) und <i>BrainStorm</i> an Tabletop und Whiteboard (Hilliges et al. 2007, S. 140).....	65

Abbildung 24: Kollaboratives Mind-Mapping am <i>DiamondTouch</i> bei <i>TableMind</i> (Altorfer 2007, S. 18) und <i>Tabletop Mind Maps</i> (Montferrat 2007, S. 1).....	66
Abbildung 25: Flexible Positionierung und Ausrichtung von Ideen bei <i>WordPlay</i> (Hunter & Maes 2008, S. 2) und am <i>cgTable</i> (Meixner 2008).....	68
Abbildung 26: Ideengenerierung und -strukturierung mit hybriden Medien und <i>Tangibles</i> am <i>AffinityTable</i> (Geyer et al. 2011, S. 480).....	69
Abbildung 27: Iterative Projektschritte beim Wasserfallmodell (nach Royce 1987, S. 330).	82
Abbildung 28: Schematische Darstellung des <i>TUIO</i> -Protokolls in einem Tabletopsystem (Tuio.org 2011).....	86
Abbildung 29: Visualisierung von Designideen für die Tabletop-Anwendung in Form eines maßstabsgetreuen Papierprototyps.....	99
Abbildung 30: Skizzen zur Visualisierung von Ideen und Relationen auf dem Tabletop: erste Ideen, 55 x 55 mm, 85 x 40 mm und 90 x 60 mm (links, v.o.n.u.) und finale Skizze 70 x 50 mm (rechts).....	100
Abbildung 31: Skizzen zur Visualisierung der Menüs am Tabletop: Allgemeines Funktionsmenü, Ø 80mm (links) und Fenstermenü, Ø 60mm (rechts).....	101
Abbildung 32: Skizzen zur Visualisierung des Speichern-Overlays am Tabletop: Erste Idee, 100x50 mm (oben), finale Skizze 180x130 mm (unten).....	102
Abbildung 33: Skizzen zur Visualisierung des Laden- bzw. Bluetooth-Overlay (am Tabletop: erste Idee für das Lade-Overlay, 125 x 95 mm (links), finale Skizze Bluetooth/Lade-Overlay, 130 x 100 mm (rechts).....	103
Abbildung 34: Skizzen zur Visualisierung des Hilfe-Overlays am Tabletop: erste Idee, 140 x 110 mm (links) und finale Darstellung, 160 x 100 mm (rechts).....	103
Abbildung 35: Skizzen zur Visualisierung von Ideen für die Benutzerschnittstelle am Android Smartphone: Tab-Layout mit <i>Activity</i> ‚Transfer‘ (links), <i>Activity</i> ‚Ideenverwaltung‘ (mittig) und <i>Activity</i> ‚Bluetooth‘ (rechts).....	104
Abbildung 36: Grafische Darstellung der <i>MT4j</i> -Architektur (MT4j.org 2011).....	107

Abbildung 37: Systemmodell der <i>Multi/Touch/Device MindMapper</i> -Anwendung.....	110
Abbildung 38: Architekturmodell für die <i>Multi/Touch/Device MindMapper</i> -Anwendung in Annäherung an das MVC-Pattern.....	112
Abbildung 39: Paketstruktur der Tabletop-Anwendung.....	113
Abbildung 40: Klassenübersicht für das Paket <code>mindMapper</code>	114
Abbildung 41: Auszug aus dem UML-Klassendiagramm des Pakets <code>model</code> mit Vererbungs- und Kompositionsbeziehungen.....	115
Abbildung 42: Auszug aus dem UML-Klassendiagramm des Pakets <code>btServer</code> mit Kompositionsbeziehungen.....	117
Abbildung 43: Arbeitsfläche der Szene <code>MindMapScene</code> am Tabletop während einer Brainstorming/Mind-Mapping Sitzung.....	119
Abbildung 44: Aufteilung der Tabletopoberfläche in vier Bereiche zur automatischen Ausrichtung von neuen Ideen und Menü-Overlays.....	120
Abbildung 45: Unistroke-Gesten zur Erstellung von Relationen (Pfeilgeste) und Löschen von Ideen und Relationen (X-Geste).....	120
Abbildung 46: Feedback bei der Erkennung von Unistroke-Gesten: a) Rot – Geste nicht erkannt, b) Grün – Geste erkannt, Aktion erfolgt c) Gelb - Geste erkannt, jedoch keine Aktion möglich (hier: bereits Elternknoten vorhanden).....	121
Abbildung 47: Auszug aus dem UML-Klassendiagramm des Pakets <code>view</code> um die Klasse <code>IdeaNodeView</code>	123
Abbildung 48: Auszug aus dem UML-Klassendiagramm des Pakets <code>view</code> um die Klassen <code>WindowMenu</code> und <code>MainMenu</code>	124
Abbildung 49: Kreisförmige Menüs für grundlegende Funktionen (links) und Fensteraktionen (rechts).....	125
Abbildung 50: Auszug aus dem UML-Klassendiagramm des Pakets <code>view</code> um die Overlay-Klassen.....	126

Abbildung 51: Overlays zum Speichern der aktuellen Arbeitsoberfläche (links) und zum Laden einer zuvor gespeicherter Ideensammlung (rechts).....	127
Abbildung 52: Hilfe-Overlay mit Erläuterung und grafischer Darstellung der möglichen Gesten (links, Originalgrafiken bereitgestellt von <i>gestureworks.com</i>) sowie Bluetooth-Overlay zur Anzeige der aktuellen Bluetoothverbindungen und eines Links zur Smartphone-Applikation als QR-Code und Text (rechts).....	128
Abbildung 53: Erläuterung der möglichen Aktionen im Hilfe-Overlay mit zusätzlicher grafischer Darstellung der einzelnen Gesten (Originalgrafiken bereitgestellt von <i>gestureworks.com</i>).....	129
Abbildung 54: Auszug aus dem UML-Klassendiagramm des Pakets <i>view</i> um die Status-MessageDialog-Klassen.....	130
Abbildung 55: Beispieldialoge beim <i>Multi/Touch/Device MindMapper</i> (Warnung vor dem Schließen der Anwendung, Fehler beim Laden Speichern).....	131
Abbildung 56: Auszug aus dem UML-Klassendiagramm der Android-Applikation des <i>Multi/Touch/Device MindMappers</i>	132
Abbildung 57: Screenshots der <i>IdeaCreationActivity</i> : Eingabe einer neuen Idee (links) und Ansicht ohne virtuelle Tastatur (rechts).....	134
Abbildung 58: Screenshots der <i>TransferActivity</i> : Normale Ansicht (links) und Transfer einer Idee über <i>Drag and Drop</i> auf einen bestimmten Bereich (rechts).	134
Abbildung 59: Screenshots der <i>BluetoothActivity</i> : Übersicht über Gerätezustände (links) sowie Bluetooth-Kopplungsanforderung (rechts).....	136
Abbildung 60: Hilfedialoge für Ideentransfer (links) und Bluetooth (rechts).....	137
Abbildung 61: Stilisierte Grafik des Tabletops zur Kennzeichnung des Transferbereichs bei der Smartphone-Anwendung: erste Version (links) und überarbeitete Version (rechts)	144

Abkürzungsverzeichnis

	Englisch	Deutsch
API	Application Programming Interface	Programmierschnittstelle
CPS	Creative Problem Solving	Kreative Problemlösung
CSCW	Computer Supported Cooperative Work / Computer Supported Collaborative Work	Rechnergestützte Gruppenarbeit
DI	Diffused Illumination	diffuse Beleuchtung
EMS	Electronic Meeting System	Elektronisches Meetingsystem
FTIR	Frustrated Total Internal Reflection	verhinderte Totalreflexion
GSS	Group Support System	Gruppenunterstützendes System
GUI	Graphical User Interface	Grafische Benutzerschnittstelle
HCI	Human Computer Interaction	Mensch-Computer-Interaktion
HD	High Definition	hochauflösend
LCD	Liquid Crystal Display	Flüssigkristallbildschirm
MDE	Multi Display Environment	
MVC	Model View Controller	
OLED	Organic Light Emitting Diode	Organische Leuchtdiode
PIN	Personal Identification Number	Persönliche Identifikationsnummer
QR Code	Quick Response Code	
SDG	Single Display Groupware	
SDK	Software Development Kit	
SVG	Scalable Vektor Graphics	Skalierbare Vektorgrafiken
TUI	Tangible User Interfaces	Tangible Benutzerschnittstellen
UTF	Universal Character Set Transformation Format	
XGA	Extended Graphics Array	

XML	Extensible Markup Language	Erweiterbare Auszeichnungssprache
WIMP	Windows, Icons, Menus and Pointers	Fenster, Ikonen, Menüs und (Maus-) Zeiger
WPF	Windows Presentation Framework	

1 Einleitung

Am Anfang einer jeden Innovation, eines neuen Produkts oder einer Dienstleistung steht eine Idee. Die Generierung einzigartiger, neuer Gedanken ist essentieller Bestandteil unserer Lebenswelt und Voraussetzung für Fortschritt jeglicher Art, sei es in Wirtschaft, Politik, Forschung und Bildung oder in anderen kulturellen Bereichen. Zur Anregung der Kreativität bei der Ideenfindung werden, insbesondere im Kontext von Firmen und Organisationen, unterschiedlichste Methoden verwendet, so etwa Brainstorming und Mind-Mapping. Die elektronische Unterstützung solcher Kreativitätstechniken steht bereits seit mehreren Jahrzehnten im Interesse der Forschung. Dort hat sich gezeigt, dass der Einsatz von Rechner-technik Einfluss auf die Effektivität bei der Arbeit in Gruppen haben kann. So können sogenannte *Gruppenunterstützende Systeme* (GSS) beispielsweise Produktionsblockaden (*production blocking*), die durch die lineare Abfolge bei der verbalen Äußerung von Ideen beim klassischen Brainstorming auftreten, durch die Möglichkeit der parallelen Kommunikation verhindern helfen. Findet jedoch dabei die Arbeit an individuellen Arbeitsstationen statt, wird wiederum die für die Kollaboration so wichtige *Face-to-Face*-Kommunikation und das Bewusstsein um die Aktionen anderer Gruppenteilnehmer, auch (*Group*) *Awareness* genannt, gefährdet. An dieser Stelle zeigt sich das Potential von Tabletopsystemen. In den letzten Jahren zunehmend ins Interesse der Forschung gerückt, ermöglichen interaktive Tische durch ihre großflächigen, in der Regel horizontal ausgerichteten, berührungsempfindlichen Oberflächen die Unterstützung von kollaborativen Arbeitsaufgaben. Aktuelle Modelle erlauben das gleichzeitige Interagieren mit Anwendungen über Touch-Gesten oder auch *Tangibles* (getaggte physikalische Objekte), wobei durch die dabei stattfindende direkte Manipulation von Bildelementen die Handlungen der anderen Gruppenteilnehmer im Sinne der *Group Awareness* in der Regel auch immer unmittelbar sichtbar sind.

Dieser Aspekt kann jedoch, je nach Zusammensetzung einer Gruppe, auch negativen Einfluss auf die Produktivität einzelner Gruppenteilnehmer haben, beispielsweise, weil diese befürchten, dass sie von anderen Personen negativ bewertet werden und deshalb ihre Handlungen direkt am Tabletop reduzieren (*evaluation apprehension*). Ein zentrales Problem stellt darüber hinaus immer noch die Eingabe von Texten an interaktiven Tischen dar. In der Regel erfolgt diese über virtuelle Tastaturen, welche durch die glatte Oberfläche des Tisches weder den ergonomischen Komfort noch das haptische Feedback traditioneller Hardware-Tastaturen bieten können und somit insbesondere für textintensive Aufga-

benstellungen nur eingeschränkt geeignet sind. Die Ergänzung einer solchen Anwendung um externe Geräte wie Smartphones, für welche im Gegensatz zu Tabletops relativ einfach taktiles Feedback in Form von Vibration und personalisierte Texteingabemethoden umgesetzt werden können, kann diesbezüglich eine alternative Herangehensweise darstellen. In Bezug auf die Angst vor Bewertung können externe Geräte zudem private Räume schaffen, in denen Benutzer ungestört arbeiten können. Damit in Zusammenhang steht ein weiterer Aspekt, der für die kollaborative Arbeit von hoher Relevanz ist: die Unterstützung von Gruppenarbeit (*teamwork*) und Individualarbeit (*taskwork*). Während Tabletops durch die gemeinsam genutzte Arbeitsfläche in der Regel eingesetzt werden, um die Gruppenarbeit zu fördern, sind Aufgaben, die einzelne Gruppenteilnehmer zur Erreichung des Gruppenziels alleine durchführen, gegebenenfalls nur eingeschränkt am Tabletop möglich – man denke hier nur an die Anzeige weiterer Medien, wie digitaler Dokumente oder Bilder, welche Teilnehmer möglicherweise im Rahmen einer Brainstormingsitzung zur Anregung von neuen Ideen heranziehen wollen, was auf einem Tabletop wertvollen Platz in Anspruch nehmen würde und gegebenenfalls gar den Arbeitsprozess der anderen Benutzer stören könnte. Der Einsatz von Smartphones kann somit an dieser Stelle zum Ziel haben, Gruppen- und Individualarbeit in einem Brainstorming / Mind-Mapping-Szenario gleichermaßen zu unterstützen. Aufbauend auf dieser Motivation und bisherigen Kenntnissen zur Umsetzung von Benutzerschnittstellen an Tabletops dokumentiert diese Arbeit die Konzeption und Implementierung der Anwendung *Multi/Touch/Device MindMapper* zur elektronischen Unterstützung von Brainstorming und Mind-Mapping durch einen Tabletop und mehrere Smartphones.

Zielsetzung dieser Arbeit ist schließlich die Konzeption und prototypische Umsetzung (*proof of concept*) einer multitouchfähigen Anwendung für den Tabletop, mit Hilfe derer Ideen beispielsweise während Brainstorming-Meetings festgehalten und zusätzlich in Form von Mind-Maps strukturiert werden können. Die Ideenerstellung wird dabei sowohl über den Multi-Touch-Tisch als auch über Smartphones, welche via Bluetooth mit dem Tisch verbunden sind, möglich sein. Um den kollaborativen Aspekt der Aufgabe nicht zu sehr in den Hintergrund zu rücken, wird im Rahmen der Arbeit zunächst lediglich die Ideenerstellung über die externen Geräte möglich sein, welche neben dem zusätzlichen Eingabekanal primär zur Unterstützung von Individualarbeit dienen. Das Strukturieren der Ideen zu Mind-Maps wird kollaborativ am Tisch erfolgen. Neben der prototypischen Umsetzung dieser Anwendung soll die Arbeit jedoch auch einen Überblick über aktuelle Forschungs- und Problemfelder leisten, die für die Entwicklung von

Benutzerschnittstellen von Tabletopsystemen von Relevanz sind, da bisher aufgrund fehlender Standards und fehlender praktischer Erfahrungen noch ein gewisser Mangel an Fachliteratur zur Anwendungsentwicklung an Tabletops zu verzeichnen ist.

Die Arbeit gliedert sich grob in zwei Teile – zu Anfang werden die jeweils relevanten theoretischen Grundlagen beleuchtet werden, um schließlich die Konzeption und technische Umsetzung der Anwendung zu beschreiben. Kapitel 2 widmet sich zunächst der Einordnung von Kreativitätstechniken wie Brainstorming und Mind-Mapping innerhalb des Forschungsbereichs der *Computer Supported Cooperative Work* (CSCW) und geht dabei insbesondere auf Faktoren ein, die bei der Produktivität von Brainstorming zur Ideengenerierung eine Rolle spielen und beleuchtet zudem deren Einfluss im Kontext von elektronisch unterstützten Gruppenprozessen. Kapitel 3 befasst sich darüber hinaus mit der für die Anwendung zentralen Tabletoptechnologie. Dabei sollen sowohl deren Entwicklung über die letzten zwanzig Jahre als auch die dabei wesentlichen Forschungs- und Problemfelder näher betrachtet werden. Aspekte des Tabletopdesign und der Touch-Interaktion sowie Fragen der Benutzerschnittstellengestaltung und der kollaborativen Nutzung von Tabletopsystemen stehen hier im Vordergrund. Das Kapitel abschließend werden schließlich bisherige Anwendungen zum Brainstorming und Mind-Mapping und dabei umgesetzte Designlösungen vorgestellt. Kapitel 4 wird, neben der Motivation für die Kombination von Tabletops und Smartphones bei der Gruppenarbeit, die Einbettung einer solchen Anwendung in einen weiteren Kontext, der Multi-Device-Interaktion, leisten. Aufbauend auf diesen theoretischen Grundlagen erfolgt schließlich in Kapitel 5 die Beschreibung der Entwicklung des Anwendungskonzepts für den *Multi/Touch/Device MindMapper*. Teil dieses Kapitels ist, neben der Spezifikation der Anforderungen an das System und Designzielen, die sich in erster Linie aus in Kapitel 3 beschriebenen Problem- und Forschungsfeldern ableiten lassen, auch die Präsentation erster Lösungen in Form eines *Low-Level*-Papierprototypen. Die Beschreibung der konkreten Implementierung des *High-Level*-Prototypen mit *MT4j* (Tabletop) und *Android* (Smartphone) folgt schließlich in Kapitel 6. Neben den technischen Grundlagen und der Beschreibung von Konzeptänderungen und Problemen im Laufe der Implementierung steht hier insbesondere die Beschreibung der Anwendungsstruktur, das heißt der einzelnen Pakete und zentraler Programmklassen, im Fokus. Kapitel 7 schließt die Arbeit mit einer Diskussion des finalen Prototyps, sowohl unter Mitbezug erster Erfahrungen aus dem Testbetrieb als auch in Hinblick auf zukünftige Möglichkeiten der Evaluation und Weiterentwicklung, ab.

2 Brainstorming und Mind-Mapping im Kontext der CSCW

Einführend werden zunächst die in der Anwendung umgesetzten Kreativitätstechniken des Brainstorming und Mind-Mapping kurz vorgestellt und sowohl in einen allgemeinen Problemlösungsprozess als auch in den Kontext der *Computer Supported Cooperative Work* (CSCW) eingebettet. Aufbauend auf einer Diskussion der Effektivität von Brainstorming bei der Ideenfindung in Gruppen und der dabei identifizierten Problembereiche wird schließlich die Motivation für eine elektronischen Unterstützung von kollaborativer Arbeit sowohl allgemein als auch speziell im Kontext von Tabletop-Anwendungen dargelegt.

2.1 Kreativitätstechniken und Computer Supported Cooperative Work

Strategien wie das Brainstorming und das Mind-Mapping lassen sich unter dem Oberbegriff ‚Kreativitätstechniken‘ zusammenfassen, welcher von Geschka wie folgt definiert wird:

Als Kreativitätstechnik ist ein Satz von Denk- und Verhaltensregeln für eine Gruppe oder ein Individuum zu verstehen, die in ihrer Gesamtwirkung das Entstehen von Ideen begünstigen und anregen. Die Regeln beinhalten sogenannte heuristische Prinzipien wie assoziieren, abstrahieren, Analogien bilden, kombinieren, variieren, usw.¹

Kreativitätstechniken kommen in der Regel als Teil eines kreativen Problemlösungsprozesses (*creative problem solving* (CPS) *process*) zur Anwendung, dessen Ziel die Entwicklung eines Konzepts oder die Lösung eines Problems sein kann.² Ein solcher Prozess kann grob in drei Phasen unterteilt werden: 1) Die Definition des Problems, 2) das Generieren von Ideen sowie schließlich 3) die Entscheidung für eine spezielle, auf den selektierten Ideen basierende Problemlösungsstrategie bzw. Konzeptentwicklung (Abbildung 1).³ Die Phase der Ideengenerierung, in welcher die jeweiligen Kreativitätstechniken zum Einsatz kommen, lässt sich dabei weiter in drei Subprozesse gliedern. An erster Stelle steht a) die eigentliche Ideengenerierung (*generation*). Während dieser Phase können Techniken wie das

¹ Geschka (2007, S. 992 f.).

² Vgl. Geschka (2007, S. 994).

³ Vgl. Tassoul & Buijs (2007, S. 19 ff.).

Brainstorming dazu beitragen, eine große Anzahl von Ideen zu generieren. Dieser unstrukturierten Sammlung von Ideen folgt schließlich b) eine Phase der Ordnung und Strukturierung. Ziel dieses Subprozesses ist es, Gemeinsamkeiten und/oder Zusammenhänge zwischen den einzelnen Ideen aufzuspüren und sichtbar zu machen (*inventorying, clustering, ordering*).⁴

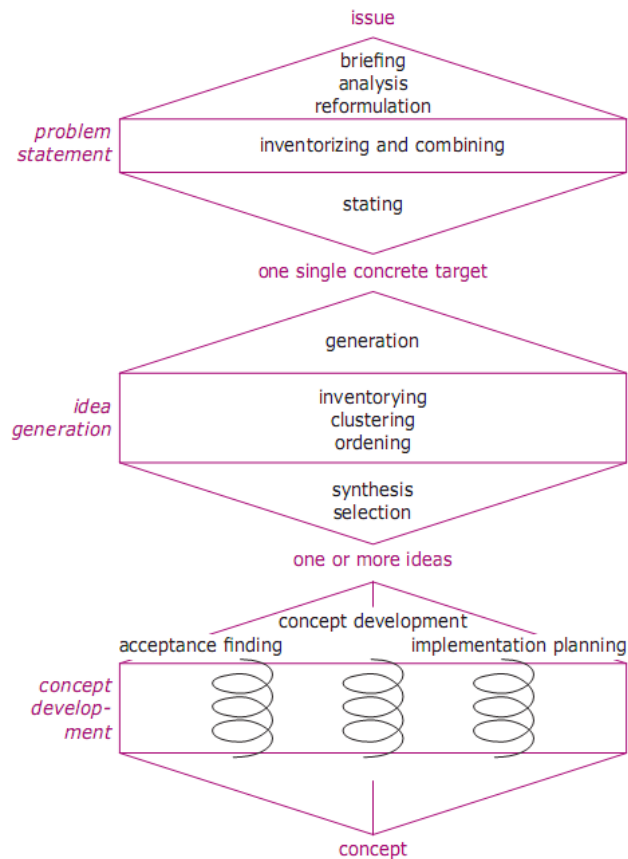


Abbildung 1: Modell des kreativen Problemlösungsprozesses nach Tassoul und Buijs (Van Boeijen & Daalhuizen 2010).

Dies kann durch die Bildung von Ähnlichkeitsmengen oder auch durch Verbindungen zwischen einzelnen Ideen geschehen. Das Mind-Mapping, welches ausgehend von einer zentralen Idee Assoziationsstrukturen aufbaut und visualisiert, umfasst somit sowohl die Phase der Ideengenerierung als auch die der Ideenstrukturierung, da gleichzeitig Ideen generiert und organisiert werden. Schließlich wird die Ideengenerierungsphase durch c) die Bewertung der gesammelten

⁴ Vgl. Tassoul & Buijs (2007, S. 20 f.).

Ideen und die Selektion jener als qualitativ überlegen erachteten Ideen, die zur Lösung des Problems beitragen sollen, abgeschlossen (*synthesis, selection*).⁵

Spätestens seit den 1980er Jahren befasst sich ein spezielles Forschungsgebiet mit der Möglichkeit der Unterstützung derartiger Prozesse durch elektronische Werkzeuge. Bei der daraus entstandenen Disziplin der *Computer Supported Cooperative Work* (CSCW) handelt es sich

um einen multidisziplinären Forschungsbereich, der sich mit dem Verstehen sozialer Interaktion sowie der Gestaltung, Implementation und Evaluierung von technischen Systemen zur Unterstützung sozialer Interaktion beschäftigt.⁶

Auf den Erkenntnissen der CSCW-Forschung aufbauende Prototypen sowie allgemeiner Systeme zur Unterstützung von Gruppenarbeit werden auch als *Groupware* oder kollaborative Software bezeichnet.⁷ Diese können weiterhin in Anwendungen unterteilt werden, die entweder verteilte oder *Face-to-Face*-Kommunikation unterstützen.⁸ Für die Anwendung von Kreativitätstechniken in der Gruppe dürften dabei vor allem letztere interessant sein, da diese Methoden von einem hohen Maß an Kollaboration profitieren. Systeme, welche die Arbeit von Gruppen speziell im Rahmen von Meetings bzw. Problemlösungsprozessen sinnvoll unterstützen sollen, werden hingegen oftmals weiter unter dem Begriff *Gruppenunterstützende Systeme* (*Group Support Systems*, GSS) oder *Elektronische Meetingssysteme* (*Electronic Meeting Systems*, EMS) zusammengefasst und bilden damit eine Untergruppe von *Groupware*.⁹ GSS können den Problemlösungsprozess unterstützen und strukturieren helfen und zudem einige Effekte, welche in Gruppensituationen auftreten und negative Auswirkungen auf die Produktivität der Gruppe haben können, reduzieren (siehe Kapitel 2.3).¹⁰ In der Folge werden somit die Kreativitätstechniken Brainstorming und Mind-Mapping näher betrachtet und die Motivation für eine elektronische Unterstützung dieser Methoden sowohl allgemein als auch unter Miteinbezug von Tabletoptechnologie dargelegt.

5 Vgl. Tassoul & Buijs (2007, S. 21).

6 Gross & Koch (2007, S. 10).

7 Vgl. Gross & Koch (2007, S. 3 ff.).

8 Vgl. Ellis, Gibbs & Rein (1991, S. 41).

9 Vgl. McFadzean (1997, S. 219), vgl. Dennis et al. (1988, S. 593), vgl. Nunamaker et al. (1991).

10 Vgl. McFadzean (1997, S. 219).

2.2 Theoretische Grundlagen zum Brainstorming und Mind-Mapping

Brainstorming

Brainstorming ist eine der bekanntesten und am häufigsten genutzten Kreativitätstechniken. Ziel dieser Methode, welche erstmals 1939 von dem in der Werbebranche tätigen Alex F. Osborn definiert wurde, ist es, während einer Sitzung so viele Ideen wie nur möglich zu generieren. Um dies zu ermöglichen, sind Kritik und Selektion von Ideen während des Brainstorming nicht erwünscht. Die Teilnehmer werden vielmehr dazu angehalten, alle Ideen spontan zu verbalisieren, ohne dabei auf deren vermeintliche Qualität zu achten.¹¹ Demnach umfasst das Brainstorming nach Osborn folgende vier Grundregeln:¹²

1. Kritik ist ausgeschlossen und sollte erst nach der Sitzung geübt werden.
2. Freie Meinungsäußerung und ungewöhnliche Ideen sind ausdrücklich erwünscht.
3. Quantität, nicht Qualität von Ideen ist das Ziel.
4. Kombination und Verbesserungen von Ideen sind erwünscht, die Teilnehmer sollen auf den Ideen anderer aufbauen.

Darüber hinaus sind laut Osborn eine gezielte Vorbereitung der Gruppe auf das Brainstorming, eine klare vorherige Definition der Arbeitsaufgabe, ein angenehmes Arbeitsklima und nicht zuletzt die Einbindung eines Gruppen- bzw. Sitzungsleiters von essentieller Bedeutung für ein produktives Gelingen eines Brainstorming.¹³ Trotz des Ausschlusses von Kritik während der Phase der Ideengenerierung spielt die Bewertung und Selektion von Ideen auch für Osborn eine zentrale Rolle im Kreativitätsprozess – sie ist jedoch nicht mehr Teil des eigentlichen Brainstorming und sollte in der Regel in einer anschließenden Sitzung durchgeführt werden.¹⁴

¹¹ Vgl. Encyclopedia of Management (2006).

¹² Vgl. Osborn (1953) zitiert nach Isaksen (1998, S. 4).

¹³ Vgl. Isaksen (1998, S. 5).

¹⁴ Vgl. Isaksen (1998, S. 4).

Mind-Mapping

Beim Mind-Mapping handelt es sich um eine Visualisierungstechnik, bei der, ausgehend von einer zentralen Idee, Assoziationen in einer radialen Struktur dargestellt werden. Dabei kann es sich um freie Assoziationen zu den jeweiligen Begriffen handeln (siehe Abbildung 2); hierarchische Strukturen – mit der zentralen Idee als Oberbegriff und davon ausgehend Abzweigungen mit spezifischeren Ideen oder Teil-Ganzes Beziehungen – sind jedoch als Sonderform des Mind-Mapping ebenfalls denkbar. Bei der Bildung solcher semantischer Netze spricht man auch von Konzept-Mapping.¹⁵ Es wird empfohlen, unterschiedliche Farben und Linienstärken zur weiteren Organisation der Mind-Map zu verwenden. Skizzen oder Bilder können ebenfalls eingesetzt werden, um die visuelle Komponente der Mind-Map zu verstärken.¹⁶

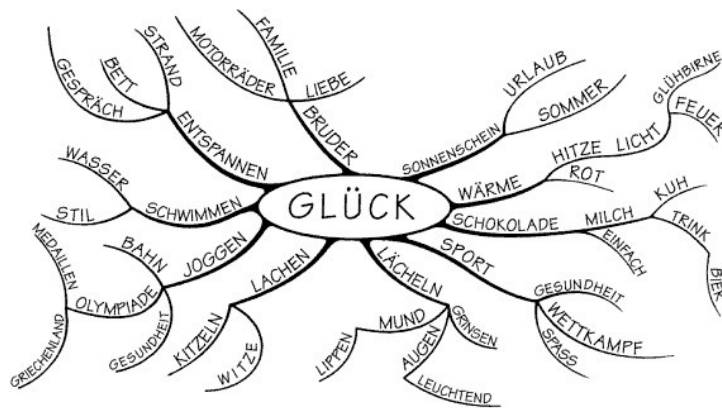


Abbildung 2: (Nicht-hierarchische) Mind-Map um den Begriff „Glück“ (Buzan & Buzan 2005, S. 81)

Mind-Mapping wurde erstmals 1974 von Tony Buzan im Rahmen seines Buches *Use Your Head* zu Lern- und Gedächtnistechniken vorgestellt. Es kann einerseits als Technik zur kreativen Ideenfindung angewandt werden, bei der durch (verschriftlichtes) Brainstorming und das spontane Bilden von Assoziationen Ideen und deren Beziehungen zueinander kreativ erschlossen werden. Andererseits wird Mind-Mapping auch zur Analyse und Strukturierung von Wissen sowie in Lehr- und Lernsituationen eingesetzt, beispielsweise beim Schreiben von Notizen während Lehrveranstaltungen, als visuelle Zusammenfassung von Texten

¹⁵ Vgl. Davies (2011, S. 282 ff.).

¹⁶ Vgl. Buzan & Buzan (2005, S. 60).

oder als Klassifikationswerkzeug.¹⁷ In einer Gruppensituation bietet das Mind-Mapping somit Potential für das kooperative Lernen, Visualisieren und Strukturieren eines gemeinsamen Problemraums.¹⁸

Die Effektivität des Mind-Mapping beruht laut Buzan schließlich auf der Annahme, dass das assoziative Denken bei der Erstellung einer Mind-Map im Gegensatz zum linearen Denken, welches beispielsweise beim Festhalten von Notizen oder Ideen in Listen zum Einsatz kommt, eher der Arbeitsweise des Gehirns entspricht und durch die Visualisierung in einer radialen Struktur zusätzlich das Erinnern und Lernen gefördert wird.¹⁹ Letzteres Argument basiert auf Allen Paivios Theorie der „dualen Codierung“ (*dual coding theory*), die davon ausgeht, dass visuelle und sprachliche Information im Gehirn getrennt verarbeitet und repräsentiert werden.²⁰ Dabei wird unter anderem angenommen, dass der kombinierte Einsatz von Sprache und visuellen Informationen positive Effekte auf das Erinnern, aber auch auf die Kreativität des Menschen hat.²¹ Somit kann in Mapping-Techniken, für welche in der Regel beide Kanäle genutzt werden, eine Möglichkeit gesehen werden, Kreativ- und Lernaufgaben positiv zu beeinflussen.²²

Einflussfaktoren auf die Produktivität des Brainstorming

Eine Vielzahl von Studien hat sich seit den 1950er Jahren mit der Frage auseinandergesetzt, wie effektiv das Brainstorming wirklich zur Generierung von Ideen eingesetzt werden kann – das heißt, ob durch diese Technik sowohl quantitativ als auch qualitativ bessere Ergebnisse ermöglicht werden. Beispielsweise hat sich im Vergleich zu Brainstorming mit Einzelpersonen herausgestellt, dass es unterschiedliche Faktoren gibt, die speziell in der Gruppensituation zum Tragen kommen und dort Prozess- bzw. Produktivitätsverluste nach sich ziehen können.²³ Folgende Effekte mit negativem Einfluss auf die Effektivität des Brainstorming in Gruppen lassen sich zusammenfassen:²⁴

- Bewertungsangst (*evaluation apprehension*) und sozialer Konformitätsdruck (*uniformity pressure*)

17 Vgl. Buzan & Buzan (2005, S. 15), vgl. Svantesson (2001, S. 13), vgl. Mandl & Fischer (2000, S. 6 f.)

18 Vgl. Mandl & Fischer (2000, S. 7 f.).

19 Vgl. Buzan & Buzan (2005, S. 46 ff., S. 59 ff.).

20 Vgl. Paivio (1979, S. 207 ff.).

21 Vgl. Paivio (1979, S. 207 ff., S. 503 ff.).

22 Vgl. Davies (2011, S. 280).

23 Vgl. Isaksen (1998, S. 5 ff.), vgl. Paulus & Yang (2009, S. 77), vgl. Taylor, Berry & Block (1958).

24 Vgl. Isaksen (1998, S. 12 f.).

- Soziales Faulenzen (*social loafing*) und Trittbrettfahrereffekt (*free riding*)
- Anpassung an die Gruppe (*social matching*) und Reduzierung der Eigenleistung aufgrund der Gefahr von Trittbrettfahrern (*sucker effect*)
- Produktionsblockaden (*production blocking*)

Die erste Gruppe von Effekten umfasst Mechanismen, die aus der Angst der Teilnehmer vor Bewertung in der Gruppe resultieren.²⁵ Obwohl die freie und kritiklose Meinungsäußerung essentieller Bestandteil einer Brainstorming Sitzung sein soll, kann es vorkommen, dass sich einzelne Teilnehmer in der Gruppe aus Angst, für ihre Äußerungen bewertet zu werden, bei der Ideengenerierung zurückhalten (*evaluation apprehension*).²⁶ Ebenso können eine gewisse Tendenz in der Gruppe oder dominierende Gruppenmitglieder dazu führen, dass sich Teilnehmer der Meinung anderer Personen beziehungsweise der Gruppe anpassen und in der Folge weniger viele oder weniger originelle Ideen äußern (*uniformity pressure*).²⁷

Des Weiteren lassen sich Szenarien beobachten, in denen einzelne Mitglieder innerhalb der Gruppe weniger Leistung zeigen. Zum einen gibt es jene, die sich auf die Leistungen der anderen Teilnehmer verlassen oder ihre eigenen Beitrag zur Ideenfindung als wertlos erachten und demzufolge weniger eigene Ideen einbringen (*social loafing, free riding*).²⁸ Zum anderen können sich ebenfalls Produktivitätsverluste zeigen, wenn sich einzelne Teilnehmer an die Leistung anderer Gruppenmitglieder anpassen (*social matching*) oder gar weniger Ideen äußern, weil sie befürchten, dass ihre Leistung von anderen, weniger stark am Ideenfindungsprozess beteiligten Gruppenmitgliedern ausgenutzt werden könnte (*sucker effect*).²⁹

Schließlich zeigen sich aufgrund der Prozessstruktur des klassischen Brainstorming erhebliche Verluste bezüglich der Produktion in der Gruppe – ist die Sitzung so angelegt, dass die Ideen verbalisiert und/oder von nur einer Person schriftlich festgehalten werden, kann es zu Produktionsblockaden kommen (*production blocking*), da einzelne Personen mit ihrer Darstellung von Ideen warten müssen, bis sie an der Reihe sind und somit eventuell einige Vorschläge gar nicht äußern können oder diese bis dahin vergessen haben.³⁰

25 Vgl. Isaksen & Gaulin (2005, S. 316).

26 Vgl. Isaksen (1998, S. 12), vgl. Diehl & Stroebe (1987, S. 498 f.).

27 Vgl. Isaksen (1998, S. 12), vgl. Taylor, Berry & Block (1958), vgl. Gryskiewicz (1984).

28 Vgl. Isaksen (1998, S. 12), vgl. Diehl & Stroebe (1987, S. 499 f.), vgl. Karau & Williams (1993).

29 Vgl. Isaksen (1998, S. 12 f.), vgl. Paulus & Dzindolet (1993), vgl. Kerr (1983).

30 Vgl. Isaksen (1998, S. 13), vgl. Diehl & Stroebe (1987, S. 498), vgl. Lamm & Trommsdorff (1973).

Als grundlegender Einflussfaktor auf die Produktivität ist somit neben der Struktur des Brainstorming insbesondere die Zusammensetzung der Gruppe zu nennen, beispielsweise die Anzahl der Teilnehmer, die Expertise der einzelnen Mitglieder, deren Bekanntheits- und Vertrautheitsgrad untereinander oder auch die Ausprägung von Hierarchien innerhalb der Gruppe.³¹ Bei großen Unterschieden bezüglich der fachlichen Kompetenz der Teilnehmer sind ebenso wie in Gruppen, deren Teilnehmer sich wenig bis gar nicht kennen, höhere Verluste durch die zuvor beschrieben sozialpsychologischen Mechanismen zu beobachten.³² Zudem kann davon ausgegangen werden, dass mit steigender Teilnehmeranzahl, insbesondere beim klassischen Brainstorming, auch die Wahrscheinlichkeit steigt, dass Produktionsblockaden auftreten.³³

Neben diesen Faktoren, die häufig als Argumente gegen die Effektivität von Gruppenarbeit und speziell Brainstorming in der Gruppe angeführt werden, kann jedoch ebenso angenommen werden, dass das dabei stattfindende Teilen einer in der Regel großen Anzahl von Ideen in einem kollaborativen Szenario kognitiv stimulierend wirkt und in Kombination mit sozialer Interaktion und Diskussion neue Assoziationen und Ideen produzieren kann, die außerhalb des Gruppenkontextes so nicht entstanden wären.³⁴

Darüber hinaus zeigen zahlreiche Studien, dass das Brainstorming restriktiveren Kreativitätstechniken, welche beispielsweise eine Selektion der Ideen bereits während deren Generation vorsehen, in Bezug auf Quantität und Qualität von Ideen überlegen ist.³⁵ Für den Erfolg der Technik ist jedoch die Einhaltung der Regeln des Brainstorming essentiell – insbesondere der Führung der Gruppe durch einen Moderator, dessen Aufgabe es ist, im Laufe einer Sitzung die Teilnehmer zur Ideengenerierung anzuregen, für eine ausgeglichene Beteiligung der einzelnen Teilnehmer sowie die Einhaltung der weiteren Regeln zu sorgen, wird in vergleichenden Studien ein nachweisbar positiver Einfluss auf die Produktivität der Gruppe während der Ideengenerierung zugeschrieben.³⁶

Die Erkenntnisse, die sich aus der Brainstorming-Forschung ziehen lassen, sind in eingeschränktem Maße auch auf das letztendlich auf dem Prinzip des Brainstorming basierenden Mind-Mapping anwendbar, wenn dieses in der Gruppe zur Ideenfindung stattfindet. Neben Produktionsblockaden bei der Ideengene-

31 Vgl. Isaksen (1998, S. 11), vgl. Mullen, Johnson & Salas (1991, S. 5 ff.).

32 Vgl. Isaksen (1998, S. 11), vgl. McFadzean (1997, S. 219), vgl. Collaros & Anderson (1969).

33 Vgl. Diehl & Stroebe (1987, S. 498), vgl. Mullen, Johnson & Salas (1991, S. 5 f.).

34 Vgl. Dugosh et al. (2000), vgl. Shih et al (2009, S. 140).

35 Vgl. Parnes & Meadow (1959), vgl. Isaksen (1998, S. 12), vgl. Isaksen & Gaulin (2005, S. 316).

36 Vgl. Isaksen (1998, S. 14 f.), vgl. Offner, Kramer & Winter (1996), vgl. Isaksen & Gaulin (2005).

rierung besteht laut Prante, Magerkurth und Streitz beim Mind-Mapping zudem die Gefahr der sogenannten Strukturierungsblockaden (*structure blocking*), wenn sich die Teilnehmer bei der Strukturierung von Ideen abwechseln müssen.³⁷

Im Zuge der Diskussion um die Effektivität des Brainstorming sind einige Varianten der Technik entstanden, die versuchen, die Prozessverluste in der Gruppensituation zu reduzieren. Das *Brainwriting* beispielsweise basiert auf dem schriftlichen Festhalten von Ideen und verzichtet auf deren verbale Äußerung, um sowohl sozialpsychologisch als auch strukturell bedingte Effekte zu verringern.³⁸ Als Weiterentwicklung des *Brainwriting* wird schließlich in der elektronischen Unterstützung von Meetings ein großes Potential für die Minimierung von Produktionsverlusten gesehen, wie in der Folge noch zu erläutern sein wird (siehe Kapitel 2.3).³⁹

Infolgedessen lässt sich festhalten, dass Brainstorming als Kreativitätstechnik bezüglich seiner Effektivität, speziell im Gruppenkontext, durchaus umstritten ist, jedoch die Regeln des Brainstorming selbst einen positiven Effekt auf die Produktivität bei der Ideenfindung in Gruppen haben können.⁴⁰ Die zuvor angeführten Effekte, die zu Produktivitätsverlusten führen können, sind nicht von der Hand zu weisen, bieten jedoch ebenso einen Ansatzpunkt für die Weiterentwicklung dieser weitverbreiteten und häufig angewandten Kreativitätstechnik. Mit der Vorstellung von GSS zur Ideengenerierung und -strukturierung sollen in der Folge Varianten der klassischen Brainstorming- und Mind-Mapping-Techniken näher beleuchtet sowie deren Potentiale im Rahmen der Motivation zu dieser Arbeit erläutert werden.

37 Vgl. Prante, Magerkurth & Streitz (2002 S. 109).

38 Vgl. VanGundy (1984).

39 Vgl. Aiken, Rebman & Vanjani (2007), vgl. Nunamaker et al. (1991).

40 Vgl. Parnes & Meadow (1959), vgl. Isaksen (1998, S. 12), vgl. Isaksen & Gaulin (2005, S. 316).

2.3 Gruppenunterstützende Systeme zur Ideengenerierung und -strukturierung

Gruppenunterstützende Systeme (GSS) werden häufig als Möglichkeit angeführt, um einige der sozialpsychologisch und strukturell bedingten Effekte, die bei der Ide-entfindung in der Gruppe auftreten, zu reduzieren und gleichzeitig die Produktivitätsgewinne, die aus der Gruppensituation sowie den jeweils verwendeten Kreativitätstechniken resultieren können, zu erhalten (Abbildung 3).⁴¹ Beispielsweise kann das Problem der Produktionsblockaden (*attention blocking, failure to remember, air time fragmentation, attenuation blocking, concentration blocking*) durch das elektronische Medium dahingehend minimiert werden, dass es parallele Kommunikation und gleichzeitige Präsentation von Ideen ermöglicht. Dadurch kann auch verhindert werden, dass wenige Personen die Sitzung durch häufigere verbale Äußerungen dominieren (*domination*).⁴² Des Weiteren erlaubt der elektronische Kanal die Anonymisierung der Ideengenerierung, welche wiederum zu einer Reduzierung der Angst vor Bewertung oder des Anpassungsdrucks an die Gruppe führen kann (*evaluation apprehension, conformance pressure*).⁴³ Schließlich ist auch das kollektive Gedächtnis (*group memory*), das durch ein digitales Speichermedium ermöglicht wird, ein Zugewinn für den Ideengenerierungsprozess.

Negative Effekte, die durch die elektronische Unterstützung von Meetings auftreten können, umfassen zum Beispiel – je nach verwendetem Medium – verzögerte Kommunikation, beispielsweise durch die Eingabe von Ideen über eine Tastatur oder langsames Feedback durch das Ausgabemedium (*slower feedback*), aus der parallelen Kommunikation resultierende Informationsüberladung (*information overload*) oder auch ein häufigeres Auftreten des zuvor erläuterten Phänomens des Sozialen Faulenzens (*social loafing*), welches teilweise mit Szenarien mit anonymen Teilnehmern in Zusammenhang gebracht wird. Ebenso wird angeführt, dass die für die Kreativität innerhalb der Gruppe so wichtige soziale Interaktion durch GSS reduziert werden kann (*socializing*).⁴⁴

41 Vgl. Gallupe, Bastianutti & Cooper (1991), vgl. McFadzean (1997), vgl. Nunamaker et al. (1991), vgl. Reinig & Shin (2002).

42 Vgl. Nunamaker et al. (1991, S. 47 ff.).

43 Vgl. Nunamaker et al. (1991, S. 49).

44 Vgl. Nunamaker et al. (1991, S. 48 f.).

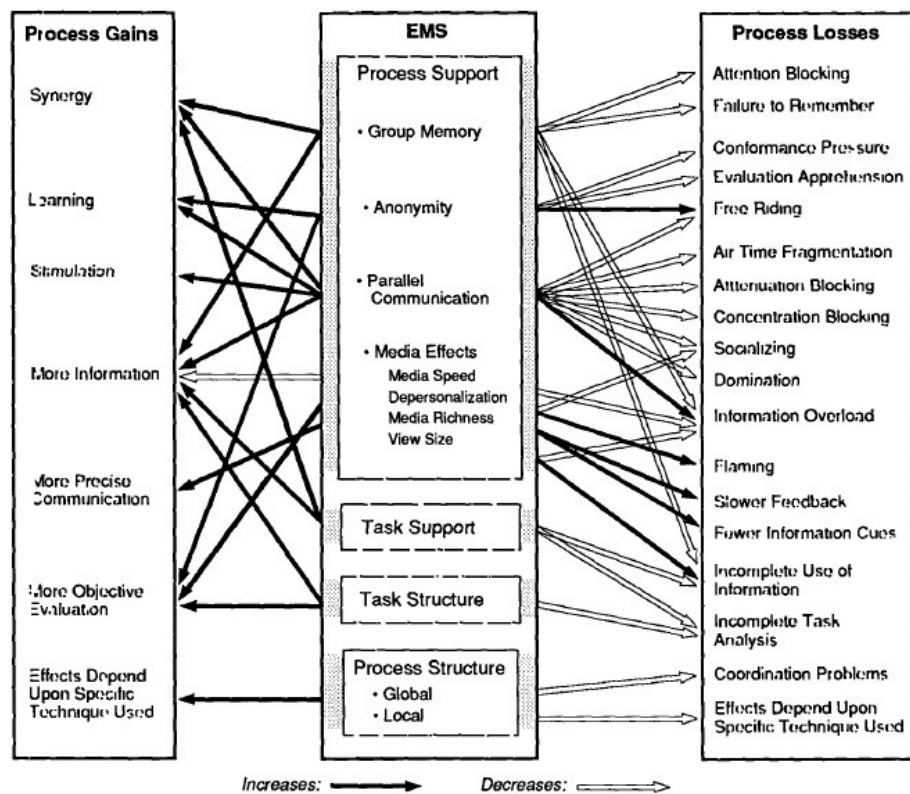


Abbildung 3: Potentielle Effekte von elektronischen Meetingsystemen (EMS) auf Prozessgewinne (*Process Gains*) und -verluste (*Process Losses*) im Gruppenkontext nach Nunamaker et al. (Nunamaker et al. 1991, S. 48)

Studien zu GSS im Meetingkontext bestätigen, dass diese zwar einerseits durch Verringerung gewisser Prozessverluste die Produktivität der Gruppe steigern können, andererseits die elektronische Unterstützung weitere Auswirkungen auf die Kollaboration haben kann. Im Allgemeinen fällt es schwer, ein klares Bild bezüglich der Effektivität und insbesondere der Zufriedenheit bei der elektronisch unterstützten Gruppenarbeit und speziell beim Brainstorming oder Mind-Mapping mit Hilfe von GSS zu zeichnen. Betrachtet man die Vielzahl der Systeme⁴⁵, die sich unter der Überschrift der GSS zusammenfassen lassen, wird klar, dass sowohl die jeweiligen Anwendungen untereinander als auch die auf ihnen aufbau-

⁴⁵ Für einen Überblick über *Gruppenunterstützende Systeme*, die zur Ideenfindung und -strukturierung eingesetzt werden können und sich weder in einen Tabletop- noch in einen Multi-Device-Kontext einbetten lassen, sei an dieser Stelle auf Forster (2010, S. 40 ff.) verwiesen. Für die Konzipierung der Anwendung relevante Systeme werden in den späteren Kapiteln zur Tabletop- und Multi-Device-Interaktion vorgestellt.

enden Studien teilweise sehr heterogen sind, und demzufolge deren Ergebnisse bezüglich der Produktivität oder auch der Zufriedenheit im Umgang mit GSS aus den verschiedensten Gründen voneinander abweichen.⁴⁶ McLeod versucht diese Erkenntnisse im Rahmen einer Metaanalyse von Studien zur elektronischen Unterstützung von Gruppenarbeit dennoch wie folgt zusammenzufassen:

It appears that GSS use leads to increased task focus and more equal participation and influence among group members. Groups produce higher quality decisions with GSS than without, but they take longer to reach those decisions. It seems that GSS may make consensus reaching more difficult, which would be consistent with the increased time to decision. The effects of GSS on user satisfaction with task processes seem to be the least clear. Field studies and some controlled laboratory studies have reported increased satisfaction, whereas other controlled laboratory studies have reported decreased satisfaction.⁴⁷

Es kann jedoch festgehalten werden, dass zahlreiche Studien zeigen, dass Gruppen, die GSS in Meetings und speziell in Brainstorming-Sitzungen einsetzen, produktiver arbeiten als Gruppen mit nicht-elektronischen Methoden, was in der Regel auf die Reduzierung der zuvor angeführten Produktivitätsverluste durch das elektronische Medium zurückgeführt wird.⁴⁸ Dass darüber hinaus der Einsatz von gruppenunterstützender Software, speziell auch im Kontext von Tabletops, Zugewinne für die kollaborative Arbeit birgt, wird in der Folge näher erläutert.

46 Vgl. McLeod (1992), vgl. Fjermestad & Hiltz (1998).

47 McLeod (1992, S. 264).

48 Vgl. McFadzean (1997, S. 219), vgl. Gallupe, Bastianutti & Cooper (1991), vgl. Reinig & Shin (2002), vgl. McLeod (1992).

2.4 Motivation für das Brainstorming und Mind-Mapping am Tabletop

Die Idee zu der im Rahmen der Arbeit umgesetzten Anwendung zum Brainstorming und Mind-Mapping am Tabletop mit Smartphones entstand im März 2011 im Rahmen einer Brainstormingsitzung am Lehrstuhl für Medieninformatik der Universität Regensburg zum Thema *Multi-Touch-Interaktion am Tabletop*. Ziel des Brainstorming war es, Ideen für zukünftige Tabletop-Anwendungen zu sammeln, die letztlich auch im Rahmen der universitären Forschung von Interesse sein können. Die dabei herausgearbeiteten Anwendungsbereiche erstreckten sich vom Software Engineering über visuelles *Information Retrieval* oder Layout- und Annotationsaufgaben, über die Unterstützung bei der Raumplanung bis hin zu *Tangible Interfaces*. Eine Vielzahl der genannten Anwendungsbeispiele ließen sich jedoch insbesondere dem Bereich ‚Meeting‘ und damit verbunden dem kollaborativen Arbeiten in Gruppen zuordnen. Ein Anwendungsfall war dabei schließlich auch das Brainstorming und Mind-Mapping am Tabletop.

Was qualifiziert nun gerade einen interaktiven Multi-Touch-Tabletop als geeignetes Interaktionsgerät zur elektronischen Unterstützung von derartigen Kreativitätstechniken? Zunächst kann angeführt werden, dass das Arbeiten von mehreren Personen an einem solchen Multi-Touch-Tisch, dessen Oberfläche sowohl als Medium für die Darstellung als auch zur Manipulation von Objekten dient, einer natürlichen Abbildung etablierter Arbeitspraktiken an traditionellen Tischen mit traditionellen Medien wie Papier entspricht.⁴⁹ Wie andere Rechnersysteme auch, bieten Tabletopsysteme darüber hinaus ebenfalls die Möglichkeit, die Ergebnisse eines kollaborativen Kreativitätsprozesses dauerhaft zu speichern und beliebig oft zu reproduzieren. Ferner sind die derzeit verfügbaren Systeme in der Regel dazu in der Lage, Eingaben mehrerer Personen, die gleichzeitig mit dem Tisch über Berührungen interagieren, zu verarbeiten (Multi-Touch-Interaktion). Bei einem Brainstorming, das durch einen Tabletop dahingehend unterstützt wird, dass die jeweiligen Ideen auf dem berührungsempfindlichen Display visualisiert und manipuliert werden können, könnte somit beispielsweise die Gefahr von Produktionsblockaden sowie bei der parallelen Arbeit an Mind-Maps die Gefahr von Strukturierungsblockaden reduziert werden.⁵⁰

⁴⁹ Vgl. Müller-Tomfelde & Fjeld (2010, S. 2).

⁵⁰ Vgl. Isaksen (1998, S. 13), vgl. Diehl & Stroebe (1987, S. 498), vgl. Prante, Magerkurth & Streitz (2002 S. 109).

Die horizontale, für alle Teilnehmer gleichzeitig benutzbare Arbeitsfläche von Tabletops fördert dabei einen demokratischen Arbeitsprozess und erschwert das Dominieren der Sitzung durch wenige Gruppenteilnehmer. Dies ist oftmals bei der Verwendung eines gemeinsamen, vertikalen Displays, beispielsweise in Form eines Desktop-Rechners oder eines digitalen Whiteboards zu beobachten. Diese unterstützen durch ihre eindeutige Orientierung, insbesondere ohne zusätzliche Eingabegeräte, in der Regel weniger viele gleichzeitig arbeitende Personen.⁵¹ Im Gegensatz zu der Verwendung von individuellen Arbeitsstationen für jeden Gruppenteilnehmer, sei es in Form von verteilter *Groupware* oder mehreren Eingabegeräte in einem Raum, unterstützt die Arbeit am Tabletop die Kollaboration dahingehend, dass die soziale Interaktion ebenso unmittelbar wie in *Face-to-Face*-Meetings erfolgen kann und darüber hinaus auch Zeigehandlungen und Referenzen auf mediale Inhalte durch die unmittelbare Nähe des Bildschirms einfacher zu interpretieren sein dürften, als beispielsweise an einem weiter entfernten Whiteboard oder gar auf mehreren Displays.⁵²

Das dabei zentrale Konzept lässt sich unter dem Begriff *Group (Activity) Awareness* zusammenfassen, welcher das Bewusstsein eines Gruppenteilnehmers um die Handlungen der anderen Mitglieder der Gruppe bezeichnet. Dieses Wissen liefert dabei einen Kontext für dessen eigene Handlungen und ist grundlegend für die Koordination von Handlungen und das Teilen von Informationen, welche wiederum Voraussetzung für eine erfolgreiche Kollaboration sind.⁵³ Wird die *Awareness* der Teilnehmer durch das System unterstützt, das heißt, sind die Aktionen der einzelnen Gruppenmitglieder im und am System für jeden unmittelbar sichtbar und von Teilnehmern kommunizierte Referenzen auf Systeminhalte leicht zu dechiffrieren, kann auch eine produktive interpersonelle Kollaboration entstehen. Reduzierte *Awareness* kann hingegen zu weniger produktiver Gruppenarbeit führen.⁵⁴ Dourish und Bellotti beschreiben hierzu die Effekte von *Awareness* auf den Kollaborationsprozess auf unterschiedlichen Ebenen:

Awareness plays a number of key roles. First, high-level awareness of the character of other's [sic] actions allows participants to structure their activities and avoid duplication of work. Second, lower-level awareness of the content

51 Vgl. Rogers & Lindley (2004, S. 16 ff.), vgl. Inkpen et al. (2005, S. 2, S. 8), vgl. Stewart, Bederson & Druin (1999, S. 288 f.), vgl. Stanton & Neale (2003).

52 Vgl. Scott, Grant & Mandryk (2003, S. 10), vgl. Rogers & Lindley (2004, S. 29 ff.).

53 Vgl. Dourish & Bellotti (1992, S. 107, S. 112).

54 Vgl. Dourish & Bellotti (1992), vgl. Gutwin & Greenberg (1998b).

of others' actions allows fine-grained shared working and synergistic group behaviour which needs to be supported by collaborative applications.⁵⁵

Dennoch zeigen sich auch Nachteile bei der Arbeit an einem Tabletop. Im Vergleich zu vertikalen Displays können weniger Teilnehmer gleichzeitig die Inhalte auf dem Bildschirm wahrnehmen, da die meisten Tische für zwei bis sechs Personen konzipiert sind und Bildelemente aus der Distanz nur schwer erkennbar sind.

Das Arbeiten mehrerer Personen an einem multitouchfähigen Tabletop ermöglicht somit einerseits die individuelle Kontrolle einzelner Teilnehmer über Inhalte, andererseits birgt ein derartiges Setting ebenso Potential bezüglich der interpersonellen Kommunikation und der *Awareness* – grundlegende Anforderungen, die für erfolgreiche Gruppenarbeit als essentiell betrachtet werden, aber in traditioneller *Groupware* oftmals miteinander konkurrieren.⁵⁶

Nach intensiver Beschäftigung mit bisherigen Forschungsarbeiten aus dem Bereich ‚Brainstorming und Mind-Mapping am Tabletop‘ und zu kollaborativem Arbeiten in diesem Kontext im Allgemeinen entstand schließlich die Idee, eine derartige Anwendung um private Displays - in diesem Fall Smartphones – zur Unterstützung der Individualarbeit im Rahmen eines Ideenfindungsszenarios zu erweitern. Die anschließenden Kapitel werden neben der Entwicklung der Tabletop-technologie, den relevanten Problem- und Forschungsfeldern und den bisherigen Forschungsarbeiten rund um das Brainstorming und Mind-Mapping am Tabletop schließlich auch die Motivation für die Erweiterung derartiger Anwendungen im Sinne einer Multi-Device Interaktion skizziert.

55 Dourish & Bellotti (1992, S. 112), Hervorhebungen im Original.

56 Vgl. Gutwin & Greenberg (1998a, S. 207).

3 Interaktive Tabletops und Multi-Touch-Interaktion

Dieses Kapitel widmet sich zunächst einem allgemeinen Überblick über die Entwicklung und Funktionsweise der Tabletoptechnologie sowie den Problem- und Forschungsfeldern, die sich in den letzten Jahren rund um die Interaktion an interaktiven, horizontalen Oberflächen herauskristallisiert haben. Diesem einführenden Teil folgt schließlich eine Vorstellung der für die vorliegende Arbeit relevanten Forschungsprojekte rund um Kreativitätstechniken am Tabletop.

3.1 Entwicklung der Tabletoptechnologie

Der Begriff ‚Tabletop‘, welcher im Englischen üblicherweise nicht viel mehr als eine Tischoberfläche bezeichnet, wurde erstmals 2001 in einem technologischen Kontext gebraucht, um überdurchschnittlich große, meist horizontal orientierte interaktive Displays bzw. interaktive Tische zu bezeichnen, welche durch Berührung, digitale Stifte oder physikalische Objekte (*Tangibles*) zu bedienen sind und somit direkte Manipulation von Benutzeroberflächenelementen erlauben.⁵⁷ Erste Projekte und Forschungsarbeiten mit Systemen, welche heute als Tabletops verstanden werden können, gehen jedoch bereits bis Anfang der 1990er Jahre zurück. Bis ab etwa 2006 die ersten Tabletops kommerziell erhältlich waren, wurde noch ein Großteil der primär in forschungsnahen Kontexten umgesetzten und untersuchten Anwendungen auf Basis jeweils individuell angefertigter Tabletophardware entwickelt. Außerhalb der jeweiligen Forschungsdomänen zeigen sich Anwendungen auf der Basis von interaktiven, horizontalen Displays spätestens seit Erscheinen des Multi-Touch-Tabletop *Microsoft Surface* im Jahr 2007 unter anderem auch vermehrt in Bildungs-, Entertainment- und Firmenkontexten.

Wenn auch die Touch-Interaktion inzwischen durch Smartphones und Tablet-PCs eine gewisse Verbreitung erfahren hat, hat sich die Tabletoptechnologie bisher nur zögerlich aus forschungsnahen Domänen herausbewegt. Dies kann sicherlich einerseits auf die noch relativ hohen Anschaffungspreise von Tabletops (die in der Regel im oberen vierstelligen bis unteren fünfstelligen Bereich liegen) sowie andererseits auf die bisher nur geringen Erfahrungen bezüglich der Langzeitnutzung derartiger Systeme und fehlender Entwicklungsstandards zurückgeführt werden. Darüber hinaus wird stellenweise ein konkreter Mangel an Anwendungsfällen in der Praxis postuliert. Das heißt die Frage, für welche Anwendungen Tabletopsysteme wirklich sinnvoll genutzt werden können,

⁵⁷ Vgl. Müller-Tomfelde & Fjeld (2010, S. 2).

scheint durchaus noch nicht zufriedenstellend beantwortet. Aufbauend auf den bisher gewonnenen Erkenntnissen zur Interaktion mit Tabletops dürfte hier durchaus noch einiges an Forschungs- und Entwicklungsarbeit nötig sein, bevor die Technologie auch in Alltagskontexten Verbreitung findet.

Dennoch hat sich in den letzten zwanzig Jahren einiges im Bereich der Tabletoptechnologie bewegt. Müller-Tomfelde und Fjeld identifizieren bezüglich der Entwicklung interaktiver Tabletops drei maßgebliche Schlüsseltransitionen, welche in der Folge anhand der Vorstellung einzelner Systembeispiele näher beleuchtet werden:⁵⁸

1. Von Forschungsprototypen zu kollaborativen Anwendungen in reellen Kontexten (1998)
2. Von Single-Touch zu Multi-Touch und *Tangible Interfaces* (2001)
3. Von Projektion zu direkter Displaytechnologie (2008)

Einer der ersten Tabletop-Prototypen wurde 1991 von Pierre David Wellner im Rahmen seiner Dissertation zur Interaktion mit Papier auf einem digitalen Tisch bei Xerox PARC entwickelt.⁵⁹ Der *DigitalDesk* besteht aus einem handelsüblichen Tisch, auf dessen leicht angewinkelte horizontale Oberfläche eine Art Benutzeroberfläche projiziert wird. Eine über dem Tisch angebracht Videokamera zeichnet die damit stattfindende Interaktion auf, wodurch Zeige- und Berührungsgesten des Benutzers (durch Finger oder Stift) registriert sowie auf den Tisch platzierte Dokumente erfasst, deren Inhalte ausgelesen und das Papier um weitere, projizierte Elemente augmentiert werden können (Abbildung 4).⁶⁰ Neben den Zeigegesten wird das Tippen auf die Oberfläche zusätzlich zur Bildverarbeitung durch die Erfassung der dabei entstehenden Laute durch ein Mikrofon erkannt.⁶¹ Für den *DigitalDesk* umgesetzte prototypische Anwendungen umfassen unter anderem eine Rechnerfunktion, eine Übersetzungsfunktion auf Wortebene sowie eine Zeichenanwendung (*PaperPaint*). Die Rechneranwendung erlaubt die Erfassung von Ziffern auf Papier durch Zeige- und Tippgesten und das Ausführen üblicher Rechenoperationen (Abbildung 4), die Übersetzungsfunktion ermöglicht auf ähnliche Weise das Übersetzen einzelner Wörter in Dokumenten auf dem Tisch (Französisch → Englisch), deren Inhalte jedoch bereits zuvor eingescannt

58 Vgl. Müller-Tomfelde & Fjeld (2010, S. 12 ff.).

59 Vgl. Wellner (1993).

60 Vgl. Wellner (1993, S. 12 ff.).

61 Vgl. Wellner (1993, S. 14).

werden müssen. *PaperPaint* bietet schließlich eine einfache Möglichkeit der Übertragung von papierbasierten Skizzen in das elektronische Medium.⁶²

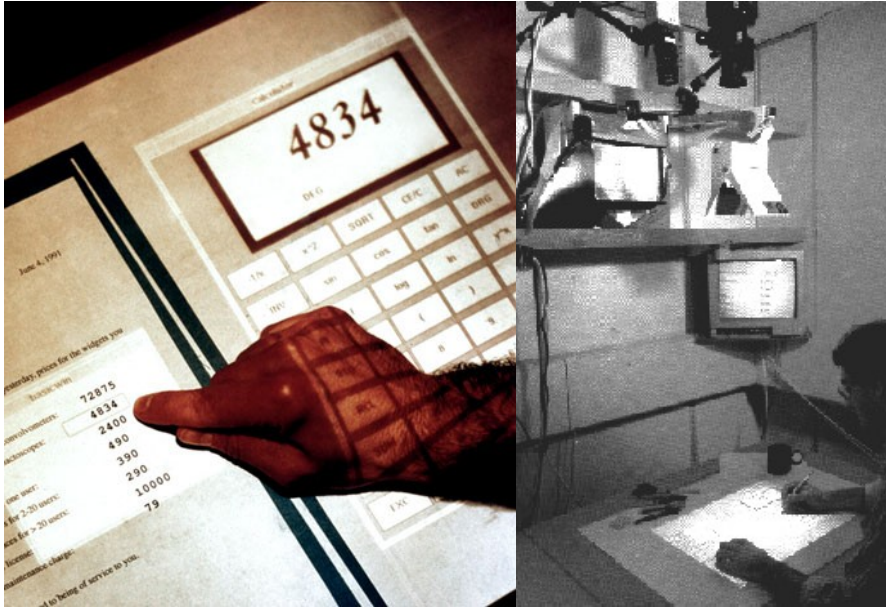


Abbildung 4: Benutzeroberfläche (Morris 2008) und Aufbau (Xerox 2011) des 1991 von Wellner entwickelten *DigitalDesk*.

Im Gegensatz zur Projektion von oben beim *DigitalDesk* wird bei dem 1995 im Rahmen der Forschungsaktivitäten rund um *Graspable* bzw. *Tangible Interfaces* von Fitzmaurice, Ishii und Buxton entwickelten *ActiveDesk* die grafische Benutzeroberfläche von unten auf die Tabletop-Oberfläche projiziert (Projektionsfläche etwa 46 Zoll Diagonale).⁶³ Darüber hinaus erlaubt der *ActiveDesk* neben Styluseingaben die „direkte Kontrolle elektronischer oder virtueller Objekte durch physikalische Artefakte, welche als Bedienelemente fungieren“⁶⁴. Motivation für diese Form von Interaktion, auch *Tangible Interaction* genannt, ist die dabei angenommene „Natürlichkeit“ der Interaktion, da hier lediglich bereits erlernte Fähigkeiten, wie das Greifen und Absetzen von physikalischen Elementen, abgerufen werden müssen.⁶⁵ Derartige Elemente in Form von Klötzchen (*Bricks*) erlauben am *ActiveDesk* beispielsweise die Auswahl oder das Bewegen und Rotieren von virtuellen Elementen in einem Zeichenprogramm. Durch die Verwendung von zwei solcher tangibler Elemente können virtuelle Objekte zusätzlich skaliert und

62 Vgl. Wellner (1993, S. 18 ff.).

63 Vgl. Fitzmaurice, Ishii & Buxton (1995).

64 Fitzmaurice, Ishii & Buxton (1995, S. 442), Übersetzung durch die Autorin.

65 Vgl. Ishii & Ullmer (1997, S. 235).

transformiert werden. Die Erkennung der Position der *Tangibles* auf dem Tisch erfolgt in diesem Fall noch über elektromagnetische Sensoren in den Klötzchen, zusätzliche Berührungseingaben sind jedoch nicht möglich.⁶⁶

Ein ähnliches Prinzip verfolgt der 1997 von Ullmer und Ishii im Rahmen des *TangibleBits* Projektes vorgestellte *metaDESK*.⁶⁷ Neben einer ebenfalls leicht angewinkelten horizontalen Oberfläche mit Rückprojektion (Auflösung 1280 x 1024 Pixel) besteht der interaktive Tisch zusätzlich aus einer „aktiven Linse“ in Form eines an einem Greifarm montierten Flachdisplays zur Darstellung zusätzlicher Inhalte (640 x 480 Pixel) in 3D sowie einer „passiven Linse“, die als transparentes *Tangible* je nach Position auf dem Tisch den jeweils selektierten Bildausschnitt in einer alternativen Ansicht projiziert (Abbildung 5).

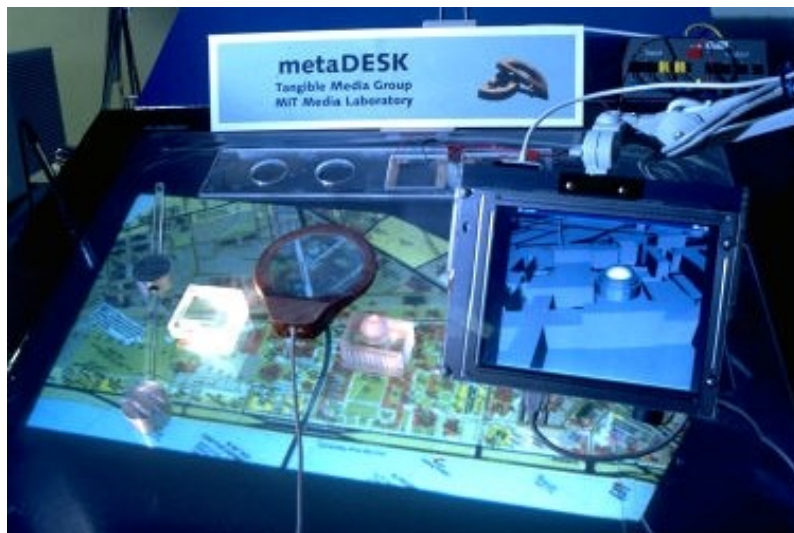


Abbildung 5: Benutzeroberfläche des *metaDESK* mit der Anwendung *Tangible Geospace* (Ullmer 2002, S. 75).

Wie bereits beim *ActiveDesk* ist hier ebenso die Interaktion mit Hilfe von physikalischen Artefakten möglich, welche durch eine Kamera im Tisch sowie elektromagnetische Feld- und Kontaktsensoren erfasst werden. Einzelne *Phicons* (*physical icons*) sind jedoch allein durch ihre Form mit unterschiedlicher Semantik belegt und können unterschiedliche Aktionen auslösen. Im Falle der Anwendung *Tangible Geospace* kann beispielsweise ein *Phicon* in Form eines bestimmten Gebäudes des *Massachusetts Institute of Technology* (MIT)-Campus auf den Tisch gesetzt wer-

⁶⁶ Vgl. Fitzmaurice, Ishii & Buxton (1995, S. 446).

⁶⁷ Vgl. Ullmer & Ishii (1997).

den, um dadurch eine entsprechende Karte anzuzeigen. Zusätzlich lässt sich die Karte durch ein oder mehrere *Phicons* manipulieren.⁶⁸

Während diese frühen, in der Regel nicht vollständig horizontal ausgerichteten Prototypen primär für die Benutzung durch eine Person intendiert sind – sei es durch Berührungsinteraktion oder über physikalische Objekte – wurde im Rahmen des Projekts *i-LAND* ab 1998 erstmals auch das kollaborative Potential der Tabletoptechnologie realisiert und somit die von Müller-Tomfelde und Fjeld identifizierte Schlüsseltransition weg von reinen Forschungsprototypen hin zu kollaborativen Anwendungen in reellen Kontexten eingeleitet.⁶⁹ Motiviert von der Vision, dass weniger der klassische Meetingraum die Kreativität der Zukunft fördere, sondern vielmehr flexible Umgebungen, die insbesondere den spontanen Informationsaustausch und informelle Begegnungen unterstützen, setzt *i-LAND* auf den Einsatz von *Roomware*, welche sich dadurch auszeichnet, dass digitale Technologie in bestehende Raumelemente wie Wände, Tische oder Stühle integriert wird, um eine „flexible und dynamische Kooperationslandschaft“⁷⁰ zu realisieren.⁷¹



Abbildung 6: Erster Prototyp des *InteracTable* 1998 (Müller-Tomfelde & Fjeld 2010, S. 8) und *i-LAND*-Roomware-Komponenten der zweiten Generation 1999 (Prante, Streitz & Tandler 2004, S. 48).

i-LAND kombiniert somit verschiedene in den Raum integrierte, teils mobile digitale Komponenten, wie ein Whiteboard (*DynaWall*), interaktive Stühle (*CommChairs*) und schließlich auch einen Tabletop mit horizontaler Oberfläche (*InteracTable*).⁷² Diese Komponenten sind über ein drahtloses Netzwerk miteinander

68 Vgl. Ullmer & Ishii (1997, S. 225 ff.).

69 Vgl. Streitz et al. (1999, S. 124 ff.).

70 Streitz et al. (1999, S. 123), Übersetzung durch die Autorin.

71 Vgl. Streitz et al. (1999, S. 123).

72 Vgl. Streitz et al. (1999, S. 123 ff.).

verbunden, wodurch digitale Objekte zwischen ihnen ausgetauscht werden können. Der erste Prototyp des *InteracTable* basiert noch auf Rückprojektionstechnik (Projektorauflösung 1024 x 768 Pixel) und visueller Erkennung der Interaktionsgesten, ein überarbeiteter Prototyp wurde ab 1999 mit einem berührungsempfindlichen Plasmadisplay ausgestattet (Abbildung 6). Beide Versionen erlauben die Interaktion mit einfachen Berührungsgesten und digitalen Stiften (Pens).⁷³

Während diese Systeme weiterhin nur die Erkennung von Single Touch-Gesten ermöglichen, findet ab etwa 2001 die zweite, für die Entwicklung der Tabletop-technologie entscheidende Transition von Single- zu Multi-Touch-Systemen statt.⁷⁴ Als einer der ersten multitouchfähigen Tabletops verfügt der 2001 von den *Mitsubishi Electronic Research Laboratories* (MERL) entwickelte *DiamondTouch* über eine weitere für die effektive kollaborative Arbeit essentielle Eigenschaft von interaktiven Oberflächen, indem er es bis zu vier Personen ermöglicht, gleichzeitig mit der Oberfläche zu interagieren.⁷⁵ Dies wird durch Antennen in der Tischoberfläche realisiert, die über den Körper des Benutzer mit Receivern in Stühlen oder Matten kapazitiv gekoppelt sind (Abbildung 7). Somit kann das System zusätzlich erkennen, durch welchen Benutzer eine Interaktion mit der Oberfläche an einer bestimmten Position erfolgt. Neben dieser Multi-Touch- und Multi-User-Inter-

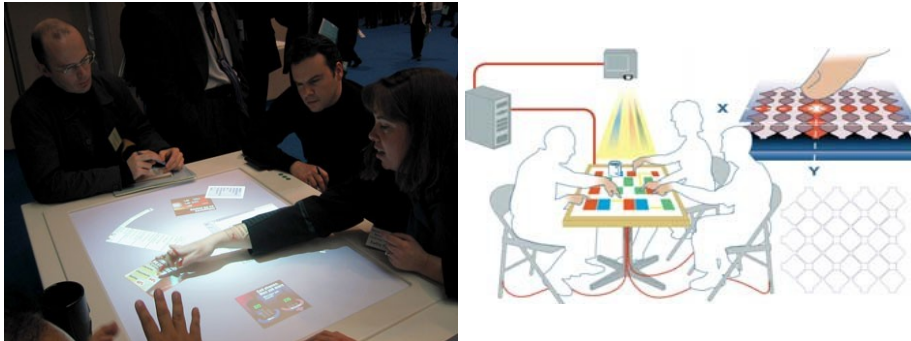


Abbildung 7: Kollaborative Interaktion am *DiamondTouch* (Morris 2008) und schematischer Aufbau (Circle Twelve Inc 2008, S. 2).

aktion sind dadurch am *DiamondTouch* auch Multi-Touch-Gesten möglich. Das heißt Berührungsgesten, die im Gegensatz zu Single-Touch-Gesten mit mehr als einem Finger ausgeführt werden müssen, können ebenfalls erkannt werden, wie beispielsweise das Vergrößern und Verkleinern von Elementen mit zwei Fingern.

⁷³ Vgl. Streitz (2010, S. vii).

⁷⁴ Vgl. Müller-Tomfelde & Fjeld (2010, S. 13 ff.).

⁷⁵ Vgl. Dietz & Leigh (2001).

Darüber hinaus wird durch die kapazitive Funktionsweise verhindert, dass auf dem Tisch abgestellte oder abgelegte Gegenstände, wie Tassen oder Schreibutensilien, versehentlich Aktionen auslösen.⁷⁶ Das *DiamondTouch*-System wurde in der Folge zusammen mit dem auf Java basierenden Entwicklungsframework *DiamondSpin* an mehrere Universitäten und Forschungseinrichtungen verteilt, wodurch eine Vielzahl von Prototypen und Studien speziell zum kollaborativen Verhalten an Tabletops entstanden sind.⁷⁷ Ab 2006 ist der *DiamondTouch* schließlich auch käuflich zu erwerben, seit 2009 wird er von *Circle Twelve Inc* in zwei Größen (32 oder 42 Zoll Bildschirmdiagonale) vertrieben.⁷⁸

Obwohl der *DiamondTouch* die gleichzeitige Interaktion durch mehrere Personen verarbeiten kann und bisher der einzige kommerziell erhältliche Tabletop ist, der die Berührungen einzelner Benutzer unterscheiden kann, hat der Systemaufbau auch entscheidende Nachteile. So kann das Erfassen von physikalischen Objekten im Sinne der *Tangible Interfaces* an dieser Art von Tabletop nicht erfolgen. Durch die Projektion der Benutzeroberfläche kommt es darüber hinaus bei der Interaktion zu Okklusion von Elementen durch Schattenwurf. Zudem wird die Anzahl der Personen, die mit dem Tisch interagieren können, von der Anzahl der Stühle bzw. Receiver bestimmt.

2005 stellt Jefferson Han mit der *Frustrated Total Internal Reflection* (FTIR)-Technik eine kostengünstige und einfach umzusetzende Alternative zu den bisher in der Regel relativ kostspieligen und komplexen Forschungsprototypen vor. FTIR ermöglicht die gleichzeitige Interaktion mehrerer Benutzer an einem Tabletop durch Multi-Touch-Interaktion und erreicht dies, im Gegensatz zum *DiamondTouch* oder zu ähnlichen Lösungen wie *SmartSkin*, nicht durch einen (kapazitiven) Touchscreen, sondern auf der Basis von optischen Verfahren.⁷⁹ Dabei wird die Benutzeroberfläche von unten auf eine Platte aus Acryl projiziert, welche von Infrarot-LEDs kantenbeleuchtet wird. Berührt ein Benutzer die Oberfläche, werden die Infrarotstrahlen an dieser Stelle abgelenkt und können von einer unter der Platte angebrachten Infrarotkamera erfasst werden (Abbildung 8).⁸⁰

76 Vgl. Dietz & Leigh (2001, S. 220).

77 Vgl. Mitsubishi Electronic Research Laboratories (2008), vgl. Müller-Tomfelde & Fjeld (2010, S. 11).

78 *Circle Twelve Inc*: <http://www.circletwelve.com>

79 Vgl. Han (2005), vgl. Rekimoto (2002).

80 Vgl. Han (2005, S. 116 f.).

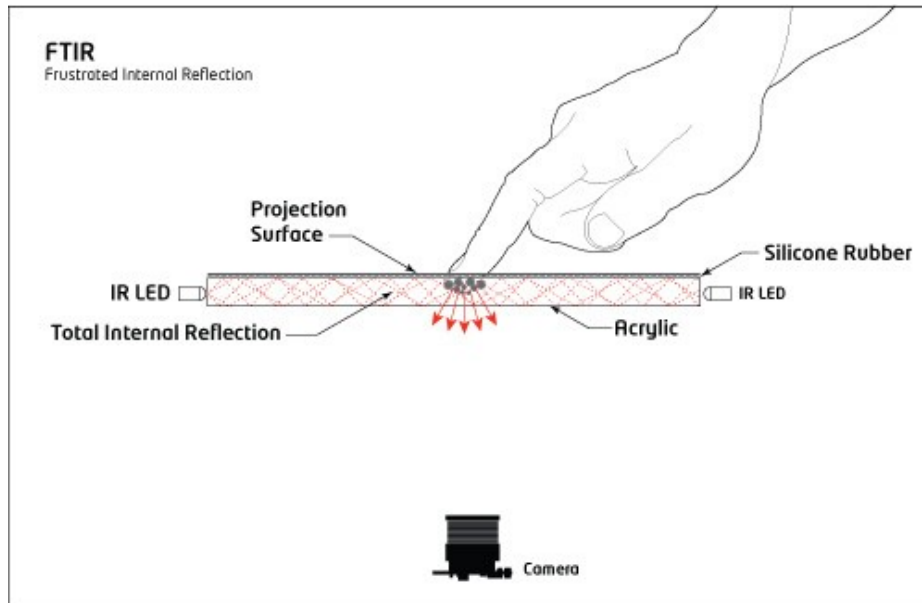


Abbildung 8: Schematischer Aufbau der *Frustrated Total Internal Reflection* (FTIR) Technik (Roth 2008)

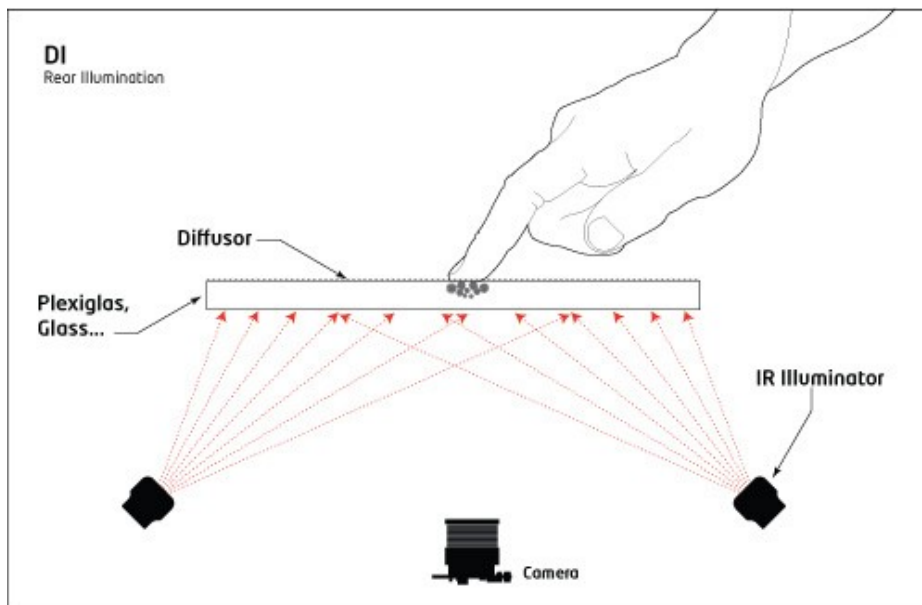


Abbildung 9: Schematischer Aufbau der *Diffused Illumination* (DI) Technik (Roth 2008)

Je nach verwendeter Erkennersoftware ist dabei die Erfassung von mehreren Dutzend Berührungspunkten gleichzeitig möglich, wodurch wiederum die Erkennung von Multi-Touch-Gesten erreicht werden kann. Die Unterscheidung einzelner Benutzer oder die Erfassung von *Tangibles* ist jedoch nicht unmittelbar möglich. Für letzteres eignet sich jedoch eine der FTIR-Technik ähnliche Methode der optischen Erfassung von Berührungspunkten, die *Diffused Illumination* (DI)-Technik, welche auch bei dem ab 2007 kommerziell erhältlichen *Microsoft Surface* zur Anwendung kommt (Abbildung 9 und 10). Dabei sitzen Projektor, Kamera und die Infrarotlichtquelle allesamt unter der Tischplatte. Durch Berührung der Bildschirmoberfläche (30 Zoll Bildschirmdiagonale, maximale Auflösung 1028 x 768 Pixel) wird das Infrarotlicht stärker reflektiert als durch das darauf angebrachte Diffusormaterial und zur Erfassung und Erkennung der Berührungspunkte durch eine oder mehrere Kameras verwendet.

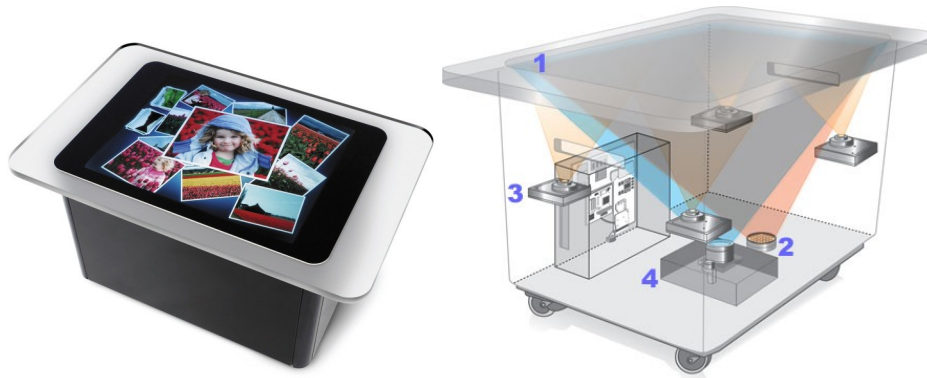


Abbildung 10: Produktbild (Microsoft 2008) und schematischer Aufbau des *Microsoft Surface* (Tzounopoulos 2007): 1) Bildschirm 2) Infrarotquelle 3) CPU 4) Projektor

Im Gegensatz zur FTIR-Technik ermöglicht dies zusätzlich die Erkennung von physikalischen Objekten über sogenannte *Tags* (Marker), die in Form von geometrischen Mustern individuelle Bitmuster repräsentieren und am Boden der Objekte angebracht werden. Somit unterstützt diese Technik erstmals sowohl Multi-Touch-Interaktion und Multi-Touch-Gesten als auch *Tangible Interaction*. Darüber hinaus ermöglicht DI – je nach verwendetem Diffusor – auch einen berührungslosen Modus, bei dem beispielsweise bereits eine Hand, die nahe der Oberfläche bewegt wird, als Berührungspunkt erkannt werden kann.⁸¹

81 Vgl. Schöning et al. (2008, S. 5), vgl. NUI Group Authors (2009, S. 13 f.).

Die dritte Schlüsseltransition der Tabletop-Technologie von der Projektion zur direkten Display-Technologie zeigt sich schließlich in den Entwicklungen der letzten Jahre. Während projektor- und kamerabasierte Systeme aufgrund ihrer massiven Bauweise oft schwer und wenig mobil sowie hochauflösende Projektoren sehr teuer sind, bietet die Kombination von Flüssigkristallbildschirmen (*Liquid Crystal Displays*, LCDs) und Multi-Touch-Erkennung Potentiale für mehr Mobilität und bessere Darstellungsqualität bei Tabletop-Systemen.

Hofer, Naeff und Kunz zeigen beispielsweise anhand ihres 2009 vorgestellten *FLATIR*-Prototyps, inwiefern sich LCDs mit der FTIR-Technik kombinieren lassen.⁸² Da die Flüssigkristallmatrix von LCDs teilweise durchlässig für Infrarotlicht ist, kann diese zwischen die kantenbeleuchtete Oberflächenplatte eines FTIR Systems und einen Infrarotsensorarray gesetzt werden um somit anhand der durch die Matrix hindurch reflektierten Infrarotstrahlen Berührungspunkte zu erfassen (Abbildung 11). Da dadurch sowohl Projektor als auch Infrarotkameras überflüssig werden, kann der dafür benötigte massive Unterbau wegfallen und das Arbeiten an Tabletops nahezu wie an einem normalen Tisch ermöglichen.⁸³

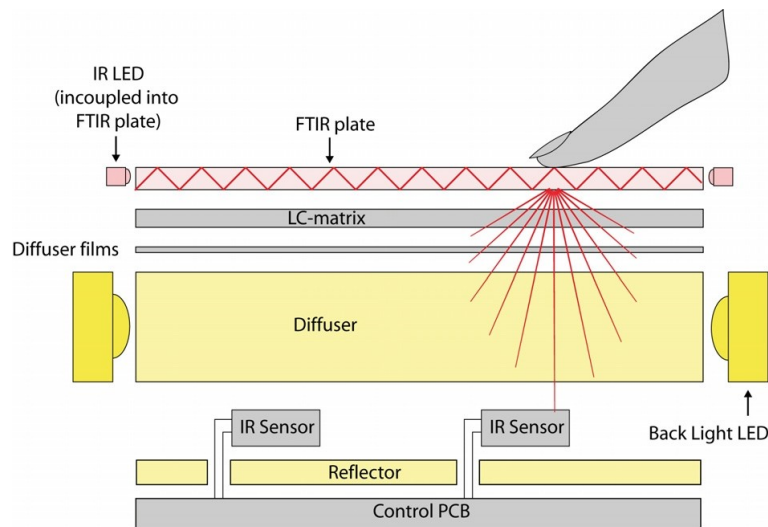


Abbildung 11: Schematischer Aufbau des *FLATIR*-Prototyps – Kombination von *FTIR*, LC-Matrix und Infrarot-Sensoren (Hofer, Naeff & Kunz 2009, S. 319)

Eine Annäherung an diese Methodik findet sich auch in bereits kommerziell erhältlichen Systemen. So verzichtet das 2010 veröffentlichte multitouchfähige

82 Vgl. Hofer, Naeff & Kunz (2009).

83 Vgl. Hofer, Naeff & Kunz (2009, S. 319).

Tabletopsystem *Evoluce ONE*, für welches auch der im Rahmen dieser Arbeit konzipierte Prototyp implementiert wurde, zugunsten eines hochauflösenden LC-Bildschirms (*Full HD LCD* mit 47 Zoll Bildschirmdiagonale, maximale Auflösung 1920 x 1080 Pixel) auf Projektorteknik (Abbildung 12). Das Erfassen von Berührungspunkten erfolgt jedoch bei der *Evoluce Integrated Through Screen Optics (ITSO)* Technologie, ähnlich wie auch bei der DI-Technik, weiterhin durch unter der Oberfläche angebrachte Kameras. Neben dem Erkennen von *Byte-Tags* zur Erfassung von physikalischen Objekten ermöglicht der *Evoluce ONE* zudem einen berührungslosen Modus.⁸⁴

Das seit 2011 erhältliche Nachfolgesystem *Evoluce TWO*, welches in Größen von 32 bis 55 Zoll Bildschirmdiagonale erhältlich ist, umfasst schließlich nur mehr einen 18,5 cm flachen LC-Bildschirm auf einem VESA-Standfuß. Multi-Touch-Gesten und Objekte auf und über der Tischoberfläche werden mit Hilfe eines über dem Tabletop angebrachten 3D-Tiefensensors erkannt (Abbildung 12).⁸⁵



Abbildung 12: *Evoluce ONE* und *Evoluce TWO* (Evoluce 2011).

Zukünftige Systeme werden sich aller Wahrscheinlichkeit nach ebenfalls weg von der Projektorteknik hin zu flachen, auf LC-Bildschirmen basierenden Aufbauten bewegen. Neben den bereits erwähnten Vorteilen gegenüber projektor- und kamerabasierten Systemen dürfte diese Technik auch günstiger in der Anschaffung sein und somit, vor allem auf lange Sicht, die Verbreitung der Tabletop-technologie auch in bisher unerschlossene Bereiche ermöglichen.

Einen Ausblick auf künftige Tabletophardware gibt das voraussichtlich Anfang 2012 erscheinende Nachfolgemodell des ersten *Microsoft Surface*, der in Koopera-

84 Vgl. Evoluce (2010).

85 Vgl. Evoluce (2011).

tion mit Samsung entwickelte *Samsung SUR40* bzw. *Microsoft Surface 2.0*.⁸⁶ Dieser multitouchfähige Tabletop umfasst einen knapp 10 cm flachen 40 Zoll Flüssigkristallbildschirm (maximale Auflösung 1920 x 1080 Pixel) mit integrierter Rechner-technik auf vier Tischbeinen, der Berührungseingaben und getaggte Objekte mit Hilfe der *PixelSense*-Technologie erkennt (Abbildung 13). Durch Abmontieren der Beine soll sich der *Samsung SUR40* auch als vertikales Display einsetzen lassen. Die *PixelSense*-Technik ähnelt der durch Hofer et al. vorgestellten *FLATIR*-Technik insofern, als hier ebenfalls eine LC-Matrix mit Infrarotsensoren kombiniert wird – die Infrarotquelle sitzt jedoch zusammen mit den Lichtquellen für das LCD unter den Sensoren zur Erfassung der Reflexionen, die durch Kontakt mit der Oberfläche entstehen.⁸⁷



Abbildung 13: *Microsoft Surface 2.0 / Samsung SUR40* (Microsoft 2011)

Die Entwicklung der Tabletop-technologie in den letzten zehn bis zwanzig Jahren skizziert somit womöglich nur die Anfänge einer zunehmenden Verbreitung von großen, horizontalen interaktiven Oberflächen. Insbesondere die rasante Entwicklung der Displaytechnologien im allgemeinen und die zunehmend standardmäßige Integration von grundlegender Multi-Touch-Funktionalität in Betriebssysteme (Beispiel *Windows 7*) und Software-Anwendungen (Beispiel *Firefox Mozilla*) dürfte einen stimulierenden Einfluss auf die Weiterentwicklung von Tabletopsystemen haben. So könnten neben der LCD-Technologie insbesondere auch *Organic Light Emitting Diodes* (OLEDs) für interaktive Oberflächen von Interesse sein, da diese noch dünnere Bildschirme, die zudem flexibel und biegsam sind, ermöglichen. Bisher wurden OLEDs jedoch nur in kleinen Displays umgesetzt.⁸⁸

86 Vgl. Samsung (2011).

87 Vgl. Samsung (2011), vgl. Cravotta (2011).

88 Vgl. Müller-Tomfelde & Fjeld (2010, S. 18).

Müller-Tomfelde und Fjeld kalkulieren für eine nachhaltige Verbreitung von Tabletops im Zuge des erst in den letzten Jahren erfolgten Technologietransfers aus dem Forschungs- in den Industriekontext weitere zehn Jahre.⁸⁹ Letztlich bleibt vor allem zu klären, für welche Anwendungen die Tabletoptechnologie in Zukunft sinnvoll einzusetzen ist. Neben der eigentlichen technischen Realisierung sind in den letzten Jahren ebenso Fragen zur Konzeption und Umsetzung von Tabletopsoftware und die Arbeit mit solchen Systemen in den Fokus der unterschiedlichen Forschungsdomänen gerückt. Die zentralen Problem- und Forschungsfelder rund um horizontale interaktive Oberflächen, welche ebenso für die Konzeption der im Rahmen der Arbeit umgesetzten Anwendung von Relevanz sind, werden in der Folge vorgestellt.

89 Vgl. Müller-Tomfelde & Fjeld (2010, S. 20).

3.2 Problem- und Forschungsfelder

Die im Bereich der Tabletopstechnologie durchgeführten Forschungsarbeiten können unterschiedlichen Forschungsdisziplinen zugeordnet werden. Als vier der wichtigsten Domänen sind hier laut Müller-Tomfelde und Fjeld die *Human Computer Interaction* (HCI), die *Computer Supported Cooperative Work* (CSCW), das *Ubiquitous Computing* und der Bereich der *Tangible User Interfaces* (TUI) zu nennen. Während sich die Disziplin der HCI primär mit der Interaktion einzelner Benutzer mit Tabletops und den dabei entstehenden Herausforderungen und neuen Interaktionsformen befasst, haben zahlreiche Arbeiten aus dem Bereich der CSCW ebenso versucht, Tabletops in Hinblick auf deren Potential für die Kollaboration in Gruppen und die soziale Interaktion zu untersuchen.⁹⁰ Weitere Anwendung finden horizontale Touch-Oberflächen zudem durch deren nahtlose Einbettung in bestehende Umgebungen im Sinne des *Ubiquitous Computing*, wie anhand der zahlreichen Projekte zu *Roomware*, die, wie im Rahmen des *i-LAND* Projektes geschehen, häufig Tabletops in umfangreiche Meetingumgebungen integrieren, ersichtlich wird.⁹¹ Nicht zuletzt haben Forschungsarbeiten aus dem Gebiet der TUI, welches sich seit den 1990er Jahren mit der physikalischen Darstellung digitaler Informationen und der Interaktion mit diesen tangiblen Elementen auseinandersetzt, die Entwicklung und Popularisierung der Tabletopstechnologie maßgeblich mit voran getrieben.⁹²

Ab etwa 2005 kann ein erhöhtes Forschungsinteresse an Tabletops und interaktiven Oberflächen, unter anderem durch eine konstant ansteigende Anzahl von Publikationen in diesem Forschungsfeld, identifiziert werden.⁹³ Seit 2006 findet mit der *International Conference on Interactive Tabletops and Surfaces* (von 2006 bis 2008 unter dem Titel *International Workshop on Horizontal Interactive Human-Computer Systems*) auch eine eigene jährliche Konferenz statt.⁹⁴ Aus der Vielzahl der Publikationen all dieser Domänen lassen sich einige zentrale Problemfelder herausarbeiten, die für die Entwicklung von Anwendungen an Tabletops von grundlegender Bedeutung sind. In der Folge werden allgemeine Aspekte, die das Tabletopdesign und die Touch-Interaktion betreffen, sowie darüber hinaus Fragen der Benutzerschnittstellengestaltung und Erkenntnisse und Herausforderun-

90 Vgl. Müller-Tomfelde & Fjeld (2010, S. 3).

91 Vgl. Weiser (1991), vgl. Prante, Streit & Tandler (2004).

92 Vgl. Ullmer (2002).

93 Vgl. Müller-Tomfelde & Fjeld (2010, S. 3 ff.).

94 *ITS 2011 - Interactive Tabletops and Surfaces*: <http://its2011.jp/>

gen bezüglich der Kollaboration an interaktiven Tabletops vorgestellt und diskutiert.

3.2.1 Tabletopdesign

Zunächst hat das allgemeine Design eines Tabletop und damit unter anderem die gewählte Anzeige- und Tracking-Technologie, der dadurch bedingte Tischaufbau sowie die Form und Dimension der Tischoberfläche Einfluss auf die damit umgesetzten Anwendungen und die daran stattfindende Arbeit.

Anzeige- und Tracking-Technologie

Die für einen Tabletop eingesetzte Technik zur Erfassung der Benutzereingaben und die verwendete Anzeigetechnik kann einen maßgeblichen Einfluss auf die Interaktion mit der Oberfläche haben. Darüber hinaus stellen sich je nach verwendeter Methodik unterschiedliche Fragen zur Mobilität, Umsetzbarkeit und Flexibilität des Systems.

Der Einsatz von Kameras zur optischen Erfassung der Benutzerinteraktion und von (rückseitigen) Projektoren zur Anzeige der Benutzeroberfläche erfordert in der Regel einen massiven Tischunterbau, der insbesondere in Situationen, in denen Personen am Tisch sitzen und nicht stehen, nachteilige Auswirkungen auf deren Bewegungsfreiheit haben kann. Zudem ist die Mobilität des Tabletops nicht zuletzt aufgrund eines hohen Gewichts eingeschränkt. Techniken, die auf Kameras und Projektoren verzichten können, oder diese über dem Tisch platzieren, erlauben schlankere und mobilere Aufbauten, die sich in Form und Optik eher an traditionelle Arbeitstische anlehnen lassen und somit die Bewegungsfreiheit der Benutzer nicht unnötig durch den Tischaufbau einschränken.⁹⁵ Ein Vorteil von optischen Erfassungsmethoden ist hingegen deren relativ einfache technische Realisierbarkeit und hohe Robustheit gegenüber Benutzereingaben. Während kapazitive oder resistive Touchscreens in der Regel industrielle Fertigung erfordern, kann ein Tabletopsystem auf Basis der FTIR- oder DI-Technik vergleichsweise einfach umgesetzt werden.⁹⁶

Ferner lassen sich in Bezug auf die verwendete Anzeigetechnik unterschiedliche Vor- und Nachteile identifizieren. Während digitale Projektoren mit Auflösungen bis zu 1024 x 768 Pixel (XGA Standard) relativ günstig in der Anschaffung sind, jedoch, insbesondere bei der Projektion von Text auf große Tischoberflä-

⁹⁵ Vgl. Schöning et al. (2010, S. 34).

⁹⁶ Vgl. Schöning et al. (2010, S. 28).

chen, zu verminderter Anzeigequalität führen können, sind höher auflösende Projektoren hingegen oftmals um ein Vielfaches teurer. Vergleichbare LCDs sind in der Regel günstiger und bieten die Vorteile der reduzierten Masse sowie besserer Helligkeitswerte, die insbesondere in hellen Umgebungen von Vorteil sind.⁹⁷

Einzelne projektorbasierte Methoden zur Anzeige der Benutzeroberfläche weisen ebenfalls diverse Vor- und Nachteile auf. So werden bei von oben projizierten Oberflächen bei der Interaktion zwangsläufig Bildelemente auf Hände und Arme projiziert und sind dadurch nicht mehr direkt auf der Oberfläche sichtbar. Durch diesen Schattenwurf verdeckte Elemente sind somit möglicherweise für andere Benutzer nur mehr schwer lokalisierbar (siehe auch Kapitel 3.2.1 und 3.2.2). Dieser Aspekt kann jedoch auch für die Entwicklung bestimmter Anwendungen und Interaktionskonzepte genutzt werden, wie beispielsweise zur Augmentierung von auf dem Tisch befindlichen Objekten oder, wie von Wu & Balakrishnan vorgeschlagen, für die Anzeige privater Informationen auf der angewinkelten Handfläche des Benutzers.⁹⁸ Für eine Projektion von oben wird hingegen selbstverständlich immer extra Hardware benötigt, die über dem Tisch montiert werden muss und nicht wie bei der Rückprojektionstechnik kompakt in das System integriert werden kann.

Infolgedessen kann festgehalten werden, dass die unterschiedlichen Anzeige- und Trackingkonzepte je nach Einsatzzweck unterschiedliche Vor- und Nachteile aufweisen, wodurch die Wahl des passenden Tabletopsystems maßgeblich von den dafür realisierten Anwendungen und diversen Kontextfaktoren abhängt. Gleichwohl ist zu erwarten, dass sich auf lange Sicht flache, auf LCDs basierende Lösungen durchsetzen.

Tischform und -dimensionen

Neben allgemeiner Aspekte zur Anzeige von Elementen und Erfassung von Eingaben kann auch die Höhe eines Tabletops sowie die Form und Dimension seiner Oberfläche Einfluss auf die Interaktion am Tisch haben. Während für länger andauernde, im Sitzen stattfindende Arbeit prinzipiell eher niedrige Tische geeignet sind, sollten für spontane oder zeitlich wenig anspruchsvolle Aufgaben, die auch im Stehen erledigt werden können oder eine höhere Mobilität der Benutzer erfordern, höhere Tische eingesetzt werden.⁹⁹ Dabei sollte jedoch laut Wallace &

97 Vgl. Schöning et al. (2010, S. 34), vgl. Koike et al. (2010, S. 131 f.).

98 Vgl. Wellner (1993), vgl. Wu & Balakrishnan (2003, S. 198 f.).

99 Vgl. Ryall et al. (2006a, S. 94 f.), vgl. Wallace & Scott (2008, S. 61).

Scott ebenso in Betracht gezogen werden, dass je nach kulturellem Kontext die für diese Aufgaben übliche Höhe eines Tisches schwanken kann.¹⁰⁰ Um mehrere Aufgabentypen und kulturelle Kontexte unterstützen zu können, wäre ein Tabletopsystem, das eine flexible Anpassung der Tischhöhe erlaubt, wünschenswert. Zum aktuellen Zeitpunkt sind jedoch weder Prototypen noch kommerzielle Systeme dieser Art bekannt. Eine Annäherung an ein derartiges System findet sich jedoch beispielsweise in Form der im *i-LAND* Projekt umgesetzten *ConnecTables*, die jeweils aus einem 13,3 Zoll großen Touchdisplay auf einem flexiblen Standfuß, welcher sowohl das Arbeiten im Sitzen als auch im Stehen erlaubt, bestehen; sowie in Form des *Evoluce ONE*-Tabletopsystems, das durch einen zusätzlichen Unterbau prinzipiell das Arbeiten sowohl im Stehen als auch im Sitzen unterstützen kann.¹⁰¹

Während eine völlig horizontal ausgerichtete Oberfläche kollaborative Arbeit mit gleichberechtigtem Einsatz aller Teilnehmer unterstützt, können angewinkelte Oberflächen die Nutzung von Anwendungen, die lediglich von einer oder wenigen Personen bedient werden, erleichtern.¹⁰² Runde Tabletops können die gleichberechtigte Arbeit bei kollaborativen Aufgaben am Tisch fördern, da diese *Face-to-Face*-Kommunikation, im Gegensatz zu quadratischen Tischen, an denen Personen auch nebeneinander arbeiten können bzw. müssen, optimal unterstützen. Kreisförmige interaktive Oberflächen wurden jedoch in der Vergangenheit nur selten in Forschungskontexten verwendet.¹⁰³

Bei der Interaktion mit dem Tabletop kann darüber hinaus beobachtet werden, dass sich Benutzer vor allem in Szenarios, welche das Sitzen am Tabletop erfordern, oftmals mit ihrem Oberkörper oder Armen auf den Tisch lehnen, weshalb Tabletopsysteme in der Regel über eine Randeinfassung verfügen. Diese kann zusätzlich zum Abstellen von Objekten, die nicht direkt auf die berührungsempfindliche Oberfläche platziert werden sollen, oder von zusätzlichen Eingabegeräten genutzt werden. Je nach Aufgabenkontext kann die optimale Breite für eine solche Randeinfassung schwanken – je mehr zusätzliche Objekte wie Papier oder externe (Eingabe-) Geräte Teil des Arbeitsprozesses am Tabletop sind, desto wichtiger wird auch eine entsprechend breite, nicht-interaktive Fläche, welche die eigentliche interaktive Oberfläche einrahmt. Laut Ryall und anderen hat sich hier bei Benutzerstudien zu kollaborativen Aufgaben eine Breite von 5-6 cm als

100 Vgl. Wallace & Scott (2008, S. 59).

101 Vgl. Tandler et al. (2001).

102 Vgl. Wallace & Scott (2008, S. 61).

103 Vgl. Jordà et al. (2007).

zu gering herausgestellt, da dadurch viele ungewollte Berührungen auf der Tischoberfläche registriert wurden (siehe auch Kapitel 3.2.2).¹⁰⁴ Cheng, Müller-Tomfelde und Ruiz schließen aus einer Studie zur Interaktion in einem Meetingszenario, dass Teilnehmer in der Regel zusätzliche Objekte wie Papier oder Laptops in die Meetingsituation mit einbringen wollen, jedoch diesem Bedürfnis angepasste, größere Randeinfassungen auch die Erreichbarkeit von Bildelementen einschränken können. Als eine zukünftige Lösung für dieses Problem wird schließlich die Möglichkeit von flexiblen, individuell dem Anwendungskontext anpassbaren, nicht-interaktiven Randbereichen vorgeschlagen.¹⁰⁵

Schließlich hat auch die Größe der Tabletopoberfläche Einfluss auf die daran stattfindende Arbeit. Größere Tische ermöglichen, dass einzelne Teilnehmer gewisse Bereiche für persönliche Individualarbeit reservieren können (siehe auch Kapitel 3.2.4), und unterstützen somit insbesondere lose gekoppelte Gruppenaufgaben, bei denen ein gewisses Maß an Individualarbeit in Ergänzung zur kollaborativen Arbeit am Tisch zu erwarten ist. Kleinere Tische sind demnach vor allem für eng gekoppelte Gruppenaufgaben bzw. Anwendungen geeignet, welche weniger Individualarbeit erfordern.¹⁰⁶

Bei der Größe des Tisches sollte auch der Aspekt der physischen Nähe der einzelnen Benutzer beachtet werden, da sich, vor allem bei kollaborativen Aufgaben, ein stark eingeschränkter persönlicher Bereich negativ auf die Arbeitssituation auswirken kann, wenn Benutzer (sozial oder kulturell motiviert) Hemmungen haben, mit dem Tisch zu interagieren, weil andere Benutzer im Weg sein könnten. Tabletops, die zur Unterstützung von kollaborativen Aufgaben eingesetzt werden, sollten somit ausreichend groß sein, um allen Teilnehmern genügend Raum zur angenehmen Interaktion mit dem System zu geben, jedoch nicht so groß, dass einzelne Elemente nicht mehr von den Benutzern erreicht werden oder Textbereiche nicht mehr gelesen werden können. Negative Effekte der Bildschirmgröße auf die Erreichbarkeit von Elementen können prinzipiell von indirekten Interaktionstechniken abgemildert werden, diese verringern jedoch möglicherweise durch das Brechen mit der direkten Manipulationsmetapher die *Awareness* innerhalb der Gruppe in Bezug auf die Aktionen anderer Teilnehmer.¹⁰⁷ Insbesondere bei großen Tabletopoberflächen zeigen sich aufgrund der Charakteristika der Touch-Interaktion zusätzliche Probleme, welche neben weiteren Fra-

¹⁰⁴ Vgl. Ryall et al. (2006a, S. 94), vgl. Cheng, Müller-Tomfelde & Ruiz (2010).

¹⁰⁵ Vgl. Cheng, Müller-Tomfelde & Ruiz (2010).

¹⁰⁶ Vgl. Wallace & Scott (2008, S. 59 f.).

¹⁰⁷ Vgl. Wallace & Scott (2008, S. 59), vgl. Ryall et al. (2004, S. 285), vgl. Nacenta et al. (2010, S. 320 f.).

gestellungen im Kontext von Berührungseingaben am Tabletop in der Folge diskutiert werden sollen.

3.2.2 Aspekte der Touch-Interaktion

Obwohl die Eingabe über Maus oder Tastatur an vielen Tabletopsystemen möglich ist, liegt der Fokus klar auf der haptischen Interaktion über direkte Berührung der Oberfläche. Aus den Charakteristika von Berührungseingaben ergeben sich einige Problemfelder, die im Rahmen der Entwicklung von Tabletops sowie für diese konzipierte Anwendungen beachtet werden müssen. Inkonsistenzen und mangelndes Feedback bei der Erkennung von Berührungen können Benutzer verunsichern und so schließlich auch negativen Einfluss auf die Benutzerzufriedenheit haben. Ein Großteil dieser Problemfelder findet sich bei Benko & Wigdor in einer Übersicht zu den Einschränkungen am Tabletop durch Berührungseingaben, welche in der Folge neben weiteren Aspekten vorgestellt werden.¹⁰⁸

Präzision, Okklusion und das *Fat-Finger-Problem*

Während die Interaktion durch Berührung oftmals als „natürlich“ und „intuitiv“ bezeichnet wird, treten im Vergleich zu traditionellen Interaktionstechniken dennoch einige Probleme auf. Je kleiner die auszuwählenden Bildelemente, desto schwieriger ist deren Auswahl durch einen Finger im Vergleich zur Auswahl durch digitale Stifte oder indirekte Zeigehandlungen mit einer Maus. Dies liegt einerseits daran, dass der Bereich des Kontaktes zwischen Finger und Oberfläche um ein vielfaches größer ist als ein Mauszeiger und durch die Nachgiebigkeit der Fingerspitze variieren kann, sowie andererseits (auszuwählende) Bildelemente bei der Auswahl durch Finger (und im Falle der Projektion von oben durch weitere Körperteile) verdeckt werden können (*Okklusion*). Zudem variiert die Größe der Finger von Benutzer zu Benutzer. Diese Problematik wird gemeinhin auch als *Fat-Finger-Problem* bezeichnet.¹⁰⁹ Dies führt insbesondere bei Benutzerschnittstellen, die ursprünglich für die Interaktion mit Tastatur und Maus entwickelt wurden, oder mobilen Geräten zu Problemen, weil viele Elemente schlicht zu klein sind oder zu nah nebeneinander stehen, um eine sichere Auswahl durch Berührung zu gewährleisten.

Für den Benutzer ist darüber hinaus oftmals auch nur schwer abzuschätzen, wo genau der Kontaktpunkt zwischen Finger und Oberfläche liegt, der die jewei-

¹⁰⁸Vgl. Benko & Wigdor (2010).

¹⁰⁹Vgl. Benko & Wigdor (2010, S. 253), vgl. Ryall et al. (2006a, S. 92).

lige Aktion auslöst. Holz und Baudisch konnten im Rahmen einer Studie zeigen, dass Benutzer den Kontaktpunkt einer Berührung auf einer horizontalen Oberfläche in der Regel falsch antizipieren, da sie diesen Punkt nur aufgrund der von oben sichtbaren geometrischen Eigenschaften des Fingers kalkulieren, beispielsweise durch Ausrichtung am horizontalen Mittelpunkt des Fingers sowie am vertikalen Mittelpunkt des Fingernagels. Der vom System erfasste Berührungspunkt entspricht jedoch in der Regel dem Mittelpunkt der Kontaktfläche (Abbildung 14).¹¹⁰

Darüber hinaus kann auch der Berührungswinkel zwischen Fingerspitze und Oberfläche, insbesondere auf horizontalen Oberflächen, schwanken – je weiter ein ausgewähltes Element von dem Benutzer entfernt ist, desto wahrscheinlicher ist es, dass der Finger weniger senkrecht auf die Oberfläche trifft und damit die Kontaktfläche vergrößert wird, was wiederum zu Fehleingaben aufgrund einer falschen Abschätzung des Kontaktpunktes durch den Benutzer führen kann.¹¹¹

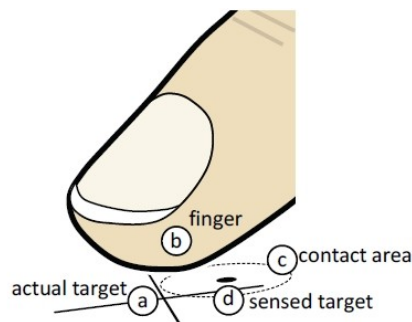


Abbildung 14: Abweichung zwischen Zielpunkt und erfasstem Berührungspunkt bei Elementauswahl. a) Tatsächlicher Zielpunkt, b) Finger, c) Kontaktfläche, d) Erkannter Berührungspunkt (Holz & Baudisch 2011, S. 2502)

Um das Problem der *Okklusion* und der unsicheren Auswahl von Elementen durch Berührung abzumildern, sollten auswählbare Bildelemente, wie zum Beispiel Buttons oder Menüelemente, eine gewisse Mindestgröße und einen Mindestabstand voneinander aufweisen. Laut einer Studie durch Wang und Ren sollten quadratische Elemente eine Mindestbreite von 11,52 mm, kreisförmige Elemente einen Radius von mindestens 5,76 mm besitzen.¹¹² *Microsofts User Experience Guidelines* zur Entwicklung von Anwendungen am Tabletopsystem *Surface*

¹¹⁰ Vgl. Holz & Baudisch (2011, S. 2502 ff.).

¹¹¹ Vgl. Forlines et al (2007, S. 652), vgl. Wang & Ren (2009, S. 1066).

¹¹² Vgl. Wang & Ren (2009, S. 1070).

empfehlen darüber hinaus eine Mindestgröße von 15 mm für auswählbare Elemente sowie einen Mindestabstand von 5 mm zwischen einzelnen Elementen.¹¹³

Ferner sind einige alternative Interaktionstechniken zur präziseren Auswahl von Elementen vorgeschlagen worden, wie beispielsweise das Versetzen des Cursors um einige Millimeter von der Kontaktfläche, das Verschieben der Anzeige oder Vergrößern des berührten Bereichs sowie Techniken, die anstelle der direkten Berührung den Mittelpunkt zwischen zwei Fingern als Cursorreferenz verwenden.¹¹⁴ Im Rahmen der Hardwareentwicklung von Tabletops sind zudem auch algorithmische Korrekturmethode denkbar, die bei der Ermittlung des Kontaktpunktes zum Einsatz kommen können, um dem von Holz und Baudisch identifizierten „mentalen Modell der Berührungsinteraktion“¹¹⁵ der Benutzer entgegenzukommen.¹¹⁶

Feedback bei Berührungseingaben

Benko und Wigdor beschreiben mit dem „touch feedback ambiguity problem“¹¹⁷ (Problem der mangelnden Eindeutigkeit von Rückmeldungen bei Berührungseingaben) eine der Berührungsinteraktion inhärente Problematik – in der Regel gibt es für den Fall, dass eine Berührung nicht oder falsch erkannt wurde, keine Rückmeldung seitens des Betriebssystems oder der Hardware, wie es beispielsweise bei Mauseingaben der Fall ist, da dort ein Cursor sichtbar ist, der sowohl über die Position der Aktion als auch deren Status Rückmeldung geben kann. Hier müssen üblicherweise auf Ebene der Anwendung in entsprechendem Umfang zusätzlich (visuelle) Rückmeldungen implementiert werden.¹¹⁸ Eine weitere Möglichkeit sind hier nicht-sprachliche auditive Rückmeldungen, die jedoch, vor allem in kollaborativen Aufgabenkontexten, für die Benutzer problematisch sein können, wenn die Eingaben nicht nach einzelnen Personen unterschieden und Fehler- oder Erfolgsmeldungen individuell an einzelne Personen gerichtet werden können.¹¹⁹ Während konsistente, zeitnahe Systemrückmeldungen grundsätzlich essentiell für die Benutzbarkeit eines Systems sind, erweisen sich Rückmeldungen über die Aktionen der einzelnen Benutzer, insbesondere auch in einem kollabo-

113 Vgl. Microsoft (2009, S. 21).

114 Vgl. Benko & Wigdor (2010, S. 253 ff.), vgl. Wang & Ren (2009, S. 1071), vgl. Wigdor & Wixon (2011, S. 75), vgl. Shen et al. (2006, S. 39).

115 Holz & Baudisch (2011, S. 2502), Übersetzung durch die Autorin.

116 Vgl. Holz & Baudisch (2011, S. 2510).

117 Benko & Wigdor (2010, S. 250).

118 Vgl. Benko & Wigdor (2010, S. 250 f.).

119 Vgl. Shen et al. (2006, S. 44).

rativen Aufgabenkontext, als grundlegend, da diese wiederum die für erfolgreiche Gruppenarbeit benötigte *Awareness* fördern. Diese Art von Feedback wird auch als *Feedthrough* bezeichnet.¹²⁰

Darüber hinaus fehlt bei Berührungseingaben, im Gegensatz zu klassischen Interaktionsgeräten wie Tastatur, Maus oder Controller, konsistentes haptisches (sowie teilweise auditives) Feedback. Zum einen kann der Auslösezeitpunkt der Berührungssaktion je nach verwendeter Technologie variieren, zum anderen erfolgt, bis auf das Berühren der in der Regel uniformen Oberfläche, keinerlei haptische Rückmeldung. Die Verwendung von physikalischen Objekten im Sinne der *Tangible Interaction* kann dieses Problem abmildern. *Tangibles* beinhalten bereits durch ihre Physikalität ein höheres Potential an taktilem Feedback und können darüber hinaus, wie der von Marquardt und anderen umgesetzte *Haptic Tabletop Puck*, zusätzliche haptische Rückmeldungen ermöglichen.¹²¹ Jedoch ist die Verwendung von physikalischen Objekten für einige Aufgaben, wie beispielsweise die Eingabe von Texten, prinzipiell ungeeignet. Die Erweiterung von digitalen Stiften um haptisches Feedback ist hier eher vorstellbar.

Ferner gibt es einige Forschungsarbeiten, die sich mit der Integration von zusätzlicher Technik in Touch-Oberflächen, die haptisches Feedback bei Berührung ermöglicht, beschäftigen.¹²² Dies kann durch den Einsatz von elektromechanischen Aktuatoren geschehen, die bei Berührung der Oberfläche Vibrationen auslösen.¹²³ Ein alternativer Ansatz, umgesetzt durch Bau und andere im Prototypen *TeslaTouch*, erlaubt haptisches Feedback durch das Prinzip der Elektrovibration: durch das Bewegen eines Fingers auf einer unter geringer Spannung stehenden, leitfähigen, isolierten Oberfläche werden Anziehungskräfte erzeugt, die vom Benutzer als Unebenheiten der Oberfläche wahrgenommen werden.¹²⁴ Diese Systeme sind jedoch für die Einzelbenutzung an kleinen Bildschirmen vorgesehen und werfen durch die auf der gesamten Oberfläche spürbaren Rückmeldungen weitere Fragen hinsichtlich eines Mehrbenutzerszenarios an großen Oberflächen wie Tabletops auf. Das von Jansen, Karrer und Borchers entwickelte *MudPad* ermöglicht im Gegensatz dazu lokalisiertes taktiles Feedback für mehrere Berührungspunkte. Der dabei eingesetzte resistive, multitouchfähige Touchscreen, dessen Anzeige von oben projiziert wird, ist um einen Elektromagnetarray und eine Schicht magnetorheologischer Flüssigkeit erweitert, um unterschiedliche Aus-

120 Vgl. Gutwin & Greenberg (1998a, S. 210).

121 Vgl. Weiss, Hollan & Borchers (2010), vgl. Marquardt et al. (2009).

122 Vgl. Benko & Wigdor (2010, S. 251 f.).

123 Vgl. Fukumoto & Sugimura (2001), vgl. Poupyrev & Maruyama (2003).

124 Vgl. Bau et al. (2010).

prägungen von Festigkeit für die Oberfläche zu ermöglichen und somit für die einzelnen Berührungen unmittelbares taktiles Feedback umzusetzen.¹²⁵

Fehlen eines *Hover*-Zustandes

Berührungsgesteuerte Systeme unterstützen oftmals nur zwei Zustände bei der Eingabe: *out of range* (nicht im zulässigen Bereich) und *in contact* (in Kontakt). Im Gegensatz zu den meisten modernen Benutzerschnittstellen fehlt ihnen somit ein dritter Zustand, üblicherweise *Hover* (dt. schweben) genannt, der beispielsweise bei der Maus durch das Bewegen des Cursors ohne das gleichzeitige Drücken einer Taste realisiert wird (Abbildung 15). Dieser zusätzliche Zustand offeriert das Potential für umfassendere Interaktionstechniken wie das Aktivieren von Vorschauen, ein sicheres Positionieren des Cursors oder auch umfangreicheres visuelles Feedback bezüglich dessen Position noch vor Auslösen einer Aktion.¹²⁶

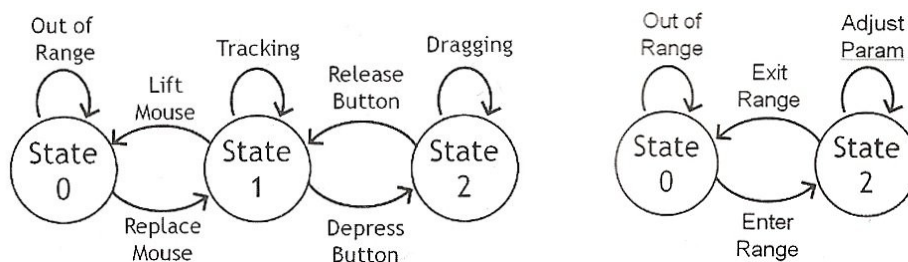


Abbildung 15: Zustandsmodelle für Eingabe via Maus (3-stufig, links) und Touch-Interaktion (2-stufig, rechts) (Benko & Wigdor 2010, S. 252 f.; adaptiert aus Buxton 1990).

Da bei Tabletopsystemen, die auf der DI-Technologie basieren, auch das Erkennen von Gesten über der Oberfläche möglich ist, wäre ein zusätzlicher *Hover*-Zustand durchaus vorstellbar. Das 2011 von Annett und anderen vorgestellte Tabletopsystem *Medusa*, das auf einem mit 138 Näherungssensoren ausgestatteten *Microsoft Surface*-Tabletop basiert, ermöglicht als eines der ersten Systeme seiner Art die Realisierung eines *Hover*-Zustandes zur Anzeige von Interaktionshilfen noch vor der Aktivierung eines Elements.¹²⁷ Alternative softwarebasierte Lösungen zur Annäherung an ein dreistufiges Eingabemodell schlagen zum Beispiel vor, das Auslösen einer Berührungsaktion auf den Zeitpunkt, zu dem der Benutzer den Finger wieder von der Oberfläche wegnimmt, zu setzen. Somit kann der

¹²⁵ Vgl. Jansen, Karrer & Borchers (2010).

¹²⁶ Vgl. Benko & Wigdor (2010, S. 252 f.).

¹²⁷ Vgl. Annett et al. (2011).

vorherige Zustand des Kontaktes mit der Oberfläche als eine Art *Hover*-Zustand verstanden werden, der unter anderem für Vorschauen und erweitertes visuelles Feedback verwendet werden kann, beispielsweise durch das Vergrößern von Elementen. Zusätzlich erlaubt dies dem Benutzer, im Gegensatz zu einem zweistufigen Zustandsmodell, das Revidieren von Selektionsentscheidungen noch vor einer Auswahl.¹²⁸

Weitere Ansätze zur präzisen Auswahl von Bildelementen beinhalten ebenfalls das Konzept eines dritten Eingabezustandes. So erfolgt bei Shen und anderen die Positionierung des Cursors durch das Platzieren und Bewegen von zwei Fingern auf der Oberfläche, wobei die Cursorposition dem Mittelpunkt zwischen beiden Fingern entspricht. Die Auswahl erfolgt analog zu einem Mausklick jedoch erst durch das Tappen mit dem Zeigefinger.¹²⁹ Gleichfalls denkbar sind Techniken wie das von Forlines, Chen und Buxton vorgestellte *Glimpse*, die diese verschiedenen Zustände durch ein unterschiedliches Ausmaß an Druck auf eine druckempfindliche Oberfläche unterscheiden.¹³⁰

Unterscheidung von beabsichtigten und unbeabsichtigten Berührungen

In Zusammenhang mit dem zweistufigen Zustandsmodell der Touch-Interaktion steht das Problem der Unterscheidung von beabsichtigten und unbeabsichtigten Berührungen der Benutzeroberfläche.¹³¹ Manche Systeme reagieren auch beim Abstellen oder Ablegen von (nicht getaggt) Objekten auf dem Tabletop genauso, als hätte eine Interaktion seitens des Benutzers stattgefunden. Dies ist in der Regel unerwünscht. Kapazitive Touchscreens (z.B. *DiamondTouch*) verhindern das fälschliche Erkennen derartiger Objekte, sofern diese nicht leitfähig sind. Ebenso vorgeschlagen wurden Benutzerschnittstellen, die nur auf Berührungen mit einer bestimmten Form reagieren.¹³² Lösungen dieser Art könnten auch die häufig auftretenden ungewollten Berührungen durch das Anlehnen von Benutzern an den Tisch, insbesondere bei schmalen Randeinfassungen, verhindern. Dieses Anlehnen führt oftmals zu ungewollten Berührungen der Oberfläche durch Arme, Handgelenke oder andere Körperteile. Ein weiteres Problem sind jedoch laut Ryall und anderen auch ungewollte Berührungen durch Zeigegesten: ein Benutzer will in einem kollaborativen Szenario beispielsweise auf ein Bildelement

128 Vgl. Benko & Wigdor (2010, S. 252 f.),

129 Vgl. Shen et al. (2006, S. 40 f.).

130 Vgl. Forlines, Chen & Buxton (2005).

131 Vgl. Benko & Wigdor (2010, S. 255).

132 Vgl. Cao et al. (2008).

ment zeigen, berührt dieses jedoch (versehentlich) und löst somit eine ungewollte Aktion aus.¹³³ Dieses Problem kann vor allem auch an Tischen auftreten, die mit optischen Verfahren wie DI arbeiten, welche Berührungen oftmals bereits kurz über der Oberfläche registrieren, worauf in der Diskussion der ersten Erfahrungen mit der im Rahmen dieser Arbeit entwickelten Anwendung noch eingegangen werden wird. Derartige Handlungen, die von O'Hara als „nicht-interaktive“¹³⁴ Handlungen bezeichnet werden, sind jedoch essentieller Bestandteil eines üblichen (kollaborativen) Szenarios an einem Tisch. Eine Einschränkung eines solchen nicht-interaktiven Verhaltens durch das Systemdesign des verwendeten Tabletop kann nicht zuletzt deshalb, weil Benutzer ungewollte Eingaben vermeiden wollen und infolgedessen ihre üblichen Interaktionshandlungen einschränken, Einfluss auf das Verhalten der Teilnehmer in einer Gruppe nehmen oder gar zu einer Verringerung interaktiver Handlungen mit dem eigentlichen System führen.¹³⁵

Ein spezielles Problem ist die Unterscheidung von Berührungseingaben nach Intention bei der handschriftlichen Eingabe von Text oder bei der Interaktion mit einer virtuellen Tastatur. Sofern das Schreiben auf dem Tisch mit Hilfe eines digitalen Stiftes erfolgt, kann dessen Eingabe von gegebenenfalls auftretenden Berührungen auf der Oberfläche durch die schreibende Hand prinzipiell differenziert werden. Soll die Eingabe jedoch auch mit dem Finger erfolgen können, erweist sich diese Unterscheidung als schwierig, was den Benutzer womöglich in eine unnatürliche Schreibhaltung zwingt, weil ansonsten fehlerhafte Eingaben durch Handgelenk oder -fläche erfolgen würden. Das selbe Problem stellt sich beim Schreiben von Texten mit virtuellen Tastaturen, wo das Tippen „in der Luft“ erfolgen muss, da die Unterarme in der Regel nicht wie an einer traditionellen Tastatur üblich auf die Oberfläche abgelegt und durch diese gestützt werden können, ohne als weitere Berührungspunkte erkannt zu werden.¹³⁶

Neben diesen Herausforderungen, die aus der Verwendung von berührungsempfindlichen Oberflächen bei Tabletops resultieren, stellen sich bei der Entwicklung von Anwendungen für Tabletopsysteme weitere Fragen in Bezug auf die Gestaltung von Benutzerschnittstellen, welche in der Folge diskutiert werden sollen.

¹³³ Vgl. Ryall et al. (2006a, S. 92).

¹³⁴ O'Hara (2010, S. 2611), Übersetzung durch die Autorin.

¹³⁵ Vgl. O'Hara (2010, S. 2612), vgl. Gerken et al. (2010).

¹³⁶ Vgl. Hinrichs et al. (2007, S. 109), vgl. Findlater, Wobbrock & Wigdor (2011, S. 2453).

3.2.3 Benutzerschnittstellengestaltung für Tabletops

Während sich für traditionelle grafische Benutzerschnittstellen (GUIs), die für klassische Eingabegeräte wie Maus und Tastaturen konzipiert und entwickelt werden, in den letzten Jahrzehnten grundlegende Standards und Patterns (Entwicklungsmuster) herausgebildet haben, erfordern innovative Benutzerschnittstellen wie Tabletops neue Ansätze und Herangehensweisen. Um das häufig angeführte Potential einer „natürlichen Eingabemodalität“¹³⁷ von Multi-Touch-Geräten wie interaktiven Tischen umfassend auszuschöpfen, muss bei der Konzeption und Entwicklung von Anwendungen vom klassischen WIMP-Paradigma (Windows, Icons, Menus and Pointers) Abstand genommen werden. Zentrale Aspekte der Schnittstellengestaltung für Tabletops, die sich aus der Forschungsliteratur der letzten Jahre herausarbeiten lassen, werden in der Folge vorgestellt.

Ausrichtung der Benutzerschnittstelle

Einer der deutlichsten Unterschiede, den Tabletops zu traditionellen Benutzerschnittstellen wie Desktoprechner aufweisen, ist ihre horizontale Ausrichtung. Ein derartiger Aufbau erlaubt eine flexible Interaktion der Benutzer mit der Oberfläche von allen Seiten. Die Beschränkung der Ausrichtung von Bildelementen auf eine Richtung, analog zu bisheriger Software, würde somit zu einer enormen Einschränkung der Interaktionsmöglichkeiten mit dem System führen. Insbesondere bei textuellen Informationen ist die korrekte Ausrichtung für die Lesbarkeit und Verständlichkeit von Inhalten von Relevanz. Anwendungen, die für den Tabletop entwickelt werden, sollten somit immer in Hinblick auf die Bedienung von allen Seiten des Tisches, ohne Fokussierung auf eine Bildschirmausrichtung, konzipiert werden.

Eine den Benutzern und der Aufgabensituation jederzeit angemessene Ausrichtung von Bildelementen zu gewährleisten, stellt jedoch eine nicht gerade triviale Herausforderung dar. Idealerweise würden für jeden Benutzer die für ihn relevanten Elemente entsprechend seiner Position am Tisch ausgerichtet, welche jedoch in der Praxis oftmals nur schwer zu ermitteln ist. Einige Systeme versuchen deshalb Annahmen darüber zu treffen, welche Elemente von welchen Benutzern oder aus welcher Perspektive verwendet werden und richten Elemente auf Basis dieser Annahmen automatisch aus, beispielsweise durch Aufteilung des Displays in bestimmte Bereiche oder durch Interpretation von Benutzerinterak-

137 Wigdor & Wixon (2011, S. 10), Übersetzung durch die Autorin.

tion.¹³⁸ Wenn die Benutzer die Ausrichtung dieser Objekte jedoch selbst nicht mehr verändern können, kann dies die Interaktion und Kollaboration am Tisch negativ beeinflussen. Scott, Grant und Mandryk empfehlen deshalb im Rahmen ihrer Guidelines zur kollaborativen Arbeit an Tabletops einen hybriden Ansatz, welcher die automatische Ausrichtung von Elementen auf Basis gewisser Annahmen über die Position des Benutzers und den Aufgabenkontext beinhaltet, jedoch den Teilnehmern ebenso die Option lässt, einzelne Bildelemente nach ihrer bevorzugten Ausrichtung zu rotieren.¹³⁹

Die Möglichkeit der individuellen Ausrichtung der Elemente durch die Benutzer ist insbesondere in kollaborativen Szenarien in den Fokus zu stellen. Neben der Tatsache, dass dadurch die Lesbarkeit und Verständlichkeit von Inhalten immer gewährleistet werden kann, hat die manuelle Ausrichtung von Elementen koordinative und kommunikative Funktion.¹⁴⁰ Kruger und andere haben im Rahmen von Benutzerstudien ermittelt, dass das Ausrichten und Bewegen von Elementen dabei hilft, Arbeitsaufgaben zu koordinieren, beispielsweise durch das Schaffen von individuellen Arbeitsbereichen (siehe auch Kapitel 3.2.4). Durch die Ausrichtung eines Elements (hin zu einem individuellen Benutzer versus hin zu den restlichen Teilnehmern) können dabei zudem Besitzverhältnisse vermittelt werden. Ebenso kann der manuellen Ausrichtung von Elementen eine kommunikative Funktion zukommen: richtet ein Benutzer ein Element zu sich hin aus, kommuniziert er damit die Absicht, alleine mit diesem Objekt zu interagieren. Orientiert er es jedoch in Richtung eines anderen Teilnehmers oder der Gruppe, fungiert die manuelle Ausrichtung des Elements als kommunikativer Akt, der idealerweise weitere Kommunikation anstößt und die Kollaboration anregt.¹⁴¹

Gesten- und Interaktionsdesign

Ein weiterer relevanter Bereich der Benutzerschnittstellengestaltung an Tabletops ist die Verwendung von interaktiven Berührungsgesten.¹⁴² Nicht zuletzt aufgrund der zunehmende Verbreitung von Touchscreens im Bereich mobiler Geräte zeigt sich in den letzten Jahren ein steigendes Interesse für diese Form der Interaktionsmodalität. Bis etwa 2001 erlauben Tabletopsysteme lediglich die Erfas-

¹³⁸ Vgl. Shen et al. (2006, S. 37), vgl. Ryall et al. (2006a, S. 93 f.).

¹³⁹ Vgl. Scott, Grant & Mandryk (2003, S. 12).

¹⁴⁰ Vgl. Kruger et al. (2003, S. 371).

¹⁴¹ Vgl. Kruger et al. (2003, S. 371 ff.).

¹⁴² Im Rahmen der Arbeit wird der Begriff der „Geste“ vereinfacht synonym zu „Berührungsgeste“ verstanden. Zur Unterscheidung von Gesten in Berührungs- und Freihandgesten siehe Saffer 2009, S. 2 ff.

sung von einzelnen Berührungseingaben, die zudem meist mit nur einer Hand bzw. einem Finger ausgeführt werden können. Die dadurch ermöglichten Gesten bilden in der Regel die gängigen Mausaktionen ab – ein Klick entspricht einem einmaligen *Tap* auf den Bildschirm, ein Doppelklick wird durch zweimaliges *Tappen* realisiert und die *Drag and Drop*-Geste (Ziehen und Loslassen) gleicht dem Auswählen, Bewegen und Loslassen von Elementen durch die Maus. In den letzten Jahren findet jedoch vermehrt die Möglichkeit der gleichzeitigen Erfassung mehrerer Berührungspunkte Verbreitung, welche wiederum Voraussetzung für sogenannte Multi-Touch-Gesten ist, die durch den gleichzeitigen Einsatz mehrerer Finger oder Hände gekennzeichnet sind und die neben zahlreichen Smartphones und Tablet-PCs inzwischen auch von der Mehrheit der aktuell verfügbaren Tabletopsystemen unterstützt werden.¹⁴³ Für komplexere Aktionen, wie das Skalieren und Rotieren von Elementen, finden sich hier häufig Mehr-Finger-Gesten, die meist den Anspruch erfüllen sollen, natürliche Bewegungsabläufe aus gewohnten Alltagskontexten abzubilden (siehe auch Kapitel 3.2.3). So wird bei der inzwischen in vielen Kontexten verwendeten Rotationsgeste das jeweilige Objekt mit zwei Fingern berührt, woraufhin eine bogenförmige Bewegung der Finger in der Rotation des Objektes resultiert, ähnlich der Vorgehensweise beim Rotieren von Papier auf einer Oberfläche. Derartige Gesten, die im Gegensatz zu „symbolischen Gesten“ (wie das Tippen auf den Bildschirm oder das Zeichnen von Symbolen und Buchstaben) keine einzelnen Aktionen auslösen, sondern eine kontinuierliche Veränderung des Systemzustandes bewirken, werden auch „Manipulationsgesten“ genannt.¹⁴⁴

Während eine derartige Steuerung von Geräten über Berührungsgesten bereits seit einigen Jahren in kommerziellen Produkten zum Einsatz kommt und sich einzelne Gesten wie jene zum Auswählen, Bewegen, Rotieren und Skalieren von Elementen bereits etabliert zu haben scheinen, stellt sich insbesondere im Kontext der Tabletoptechnologie die Frage, ob diese Gesten in der Tat immer passend für die ihnen zugewiesenen Aufgaben sind, ob sie von den Benutzern verstanden werden und – vor allem aus der Perspektive einer Langzeitnutzung – einfach in der Durchführung sind. Wobbrock, Morris und Wilson haben hierzu im Rahmen einer Studie versucht, ein Gestenset für gängige Aktionen an horizontalen Oberflächen auf der Basis von Benutzervorschlägen zusammenzustellen. Dafür wurde die Wirkung einer Geste (Element ist ausgewählt, Element wurde vergrößert etc.) präsentiert und die zwanzig Teilnehmer der Studie nach ihrer

¹⁴³ Vgl. Müller-Tomfelde & Fjeld (2010, S. 13).

¹⁴⁴ Vgl. Jetter, Gerken & Reiterer (2010, S. 2).

Präferenz bezüglich der diesem Zustand vorangehenden Ursache, das heißt der eigentlichen Geste, befragt. Dabei zeigte sich, dass Benutzer meist einfache Gesten präferieren, die mit einem Finger und einer Hand durchgeführt werden können sowie Gesten, die Handlungen in der realen Welt entsprechen.¹⁴⁵ Auch wenn auf Basis der Benutzervorschläge ein allgemeines Gestenset abgeleitet wurde, kann diesbezüglich ebenso argumentiert werden, dass die Varianz der dabei von den Benutzern vorgeschlagenen Gesten zu hoch ist, um die Fokussierung auf bestimmte Gesten zu begründen. Abbildung 16 zeigt hierzu den Anteil übereinstimmender Gesten für einzelne Aktionen in Prozent.¹⁴⁶

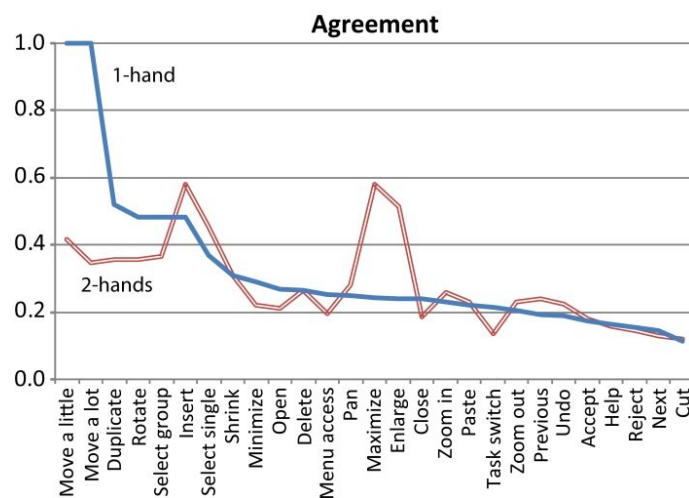


Abbildung 16: Anteil der Übereinstimmung bei benutzergenerierten Gesten für 27 unterschiedliche Aktionen am Tabletop in Prozent (Wobbrock, Morris & Wilson 2009 S. 1088).

Widgor und Wixon argumentieren an dieser Stelle, dass diese Ergebnisse zeigen, dass eine auf der Interaktion mit Gesten basierende Benutzerschnittstelle nicht rein auf der Basis von system- und anwendungsunabhängigen Benutzervorschlägen entwickelt werden kann. So etwas wie eine konsensfähige, „natürliche Geste“¹⁴⁷ für eine bestimmte Aktion gäbe es (bis auf einfache Manipulationen von Objekten, wie Bewegen) nicht, da diese auch immer vom individuellen Anwendungskontext abhänge. Hinrichs und Carpendale führen darüber hinaus, aufgrund der Ergebnisse einer ähnlichen Studie, die Relevanz allgemeiner Kontextfaktoren, wie des jeweiligen Interaktionskontextes sowie des sozialen Kon-

¹⁴⁵ Vgl. Wobbrock, Morris & Wilson (2009, S. 1087 ff.).

¹⁴⁶ Vgl. Widgor & Wixon (2011, S. 208 f.).

¹⁴⁷ Widgor & Wixon (2011, S. 209), Übersetzung durch die Autorin.

textes, für die Präferenz für bestimmte Gesten an.¹⁴⁸ Zudem übernehmen Benutzer, die mit einer neuen Technologie konfrontiert werden, häufig lediglich ihr bisheriges Wissen in die neue Domäne, ohne die Möglichkeiten, die innovative Interaktionstechniken bereitstellen, zu nutzen.¹⁴⁹

Wigdor und Wixon sehen deshalb in erster Linie den Entwickler verantwortlich für das Design eines für eine bestimmte Anwendung geeigneten Gesteninventars, wobei verallgemeinerte Gestensets, ähnlich des bei Morris, Wobbrock und Wilson abgeleiteten Inventars, sowie bereits etablierte Gesten für einfache Aktionen durchaus Hilfestellung geben können. Der Miteinbezug der Nutzer erfolgt für sie jedoch erst im Nachhinein im Rahmen der (inkrementellen) Evaluierung und Anpassung einer konkreten Anwendung. Ziel sei vielmehr, die *Affordanz*¹⁵⁰ der möglichen Gesten insofern zu steigern, als die digital manipulierbaren Objekte durch Form und Verhalten bereits genügend Hinweise auf mögliche Gesten geben.¹⁵¹ Dabei stünde nicht die Vorhersehbarkeit von Gesten ohne jegliche Hilfestellung im Fokus, sondern das langfristige Ziel der Erlernbarkeit und sicheren Durchführung einer Gestensprache durch die Kombination solcher *Affordanzen* mit entsprechendem Feedback.

Demzufolge lässt sich festhalten, dass die Art und Weise, wie angemessene Gesten für berührungsempfindliche Oberflächen konzipiert und umgesetzt werden können und wie stark der Benutzer in diesen Prozess miteinbezogen werden soll, durchaus noch kontrovers diskutiert wird. Auch wenn die Abbildung von Aktionen auf Berührungsgesten stark von der jeweiligen Anwendung abhängt, können verallgemeinerte Gestensets in Ermangelung allgemein anwendbarer Design-Patterns (Entwicklungsmustern) grundlegende Hinweise auf passende Metaphern geben.

Texteingabemethoden

Eine weitere Herausforderung stellt die Eingabe von Texten an Tabletops dar. Werden keine externen Geräte verwendet, erfolgt diese in der Regel mit Hilfe einer virtuellen Tastatur, die auf (einzelne) Berührungseingaben reagiert und üblicherweise in Form und Dimension einer traditionellen Tastatur entspricht. Das

¹⁴⁸ Vgl. Hinrichs & Carpendale (2011, S. 3031 f.).

¹⁴⁹ Vgl. Wigdor & Wixon (2011, S. 152 f., S. 209).

¹⁵⁰ *Affordance*, dt. *Affordanz*: „Beschreibung des durch visuelle/physische Eigenschaften eines Objektes kommunizierten Aufforderungscharakters im Sinne von Hinweisen zur Nutzung und Interaktion mit dem Objekt“, Usability in Germany (2011).

¹⁵¹ Vgl. Wigdor & Wixon (2011, S. 209).

Schreiben von Texten ist damit jedoch oftmals umständlich und langsam. Handschriftliche Eingaben und symbolbasierte Eingabemethoden sind häufig ebenfalls am Tabletop möglich, jedoch insbesondere für die Eingabe längerer Texte nicht zu empfehlen.¹⁵² Ryall und andere sehen dies als äußerst problematisch und argumentieren für den Einsatz externer Geräte speziell zur Eingabe längerer Texte:

Bare fingers are insufficient for text input. Our experiences with the table have shown that text input is particularly challenging. Providing virtual keyboards on the tabletop has proved a feasible, but tedious, solution. Graffiti-style input using "finger-ink" [...] is also not a practical solution for large amounts of text entry, because people draw large, clumsy shapes with fingers. Auxiliary input sources, such as wireless keyboards or PDAs with styli might be a promising solution to incorporating serious text-entry into tabletop groupware.¹⁵³

Externe Geräte zur Eingabe von Text, wie beispielsweise traditionelle Tastaturen, verfügen im Gegensatz zum Tabletop in der Regel über haptisches Feedback. Deren Einsatz erfordert jedoch, vor allem bei mehreren Personen, zusätzlichen Platz am Tabletop und kann sich gegebenenfalls störend auf das Arbeiten am Tisch auswirken, wenn häufig zwischen normaler Texteingabe und Berührunginteraktion gewechselt werden muss. Mobile Geräte wie PDAs oder Smartphones benötigen weniger Platz und bieten unter anderem das Potential der Personalisierung von Texteingabemöglichkeiten, wie im Rahmen der Argumentation für den Einsatz von Smartphones an Tabletops in Kapitel 4.2 noch weiter erläutert werden wird. Die Eingabe von Text über Spracherkennungssoftware stellt eine weitere Option dar, jedoch ist zu vermuten, dass die ständige sprachliche Interaktion mit dem System durch mehrere Personen, speziell in einem kollaborativen Szenario, eher störend ist.¹⁵⁴ Auch hier zeigt sich erneut, dass eine angemessene Form der Eingabe von der Art der Anwendung und weiteren Kontextfaktoren abhängig ist. Grundsätzlich lässt sich jedoch festhalten, dass Anwendungen, die die Eingabe von umfangreichen Texten erfordern, zum aktuellen Zeitpunkt prinzipiell eher ungeeignet für die Umsetzung an einem Tabletop sind. Letztlich fehlen an dieser Stelle bisher aussagekräftige Studien, die unterschiedliche Eingabemethoden bezüglich ihrer Effektivität, Effizienz und Benutzbarkeit vergleichen. Ein weiteres Forschungsdesiderat stellt die Verbesserung und Anpassung von virtuellen Tastaturen für Tabletops dar, die neben den offensichtlichen Problemen

152 Vgl. Hinrichs et al. (2007, S. 108 ff.).

153 Ryall et al. (2006a, S. 94).

154 Vgl. Hinrichs et al. (2007, S. 107 f.).

auch Vorteile hinsichtlich Flexibilität, Individualisierung und Reproduzierbarkeit haben. Findlater, Wobbrock und Wigdor haben hierzu erst 2011 eine der ersten Studien zum Schreibverhalten auf horizontalen Oberflächen durchgeführt. Die Ergebnisse zeigen, dass die üblicherweise rechteckig geformten virtuellen Tastaturen mit uniformem Tastenaufbau dem beobachteten Tippverhalten auf flachen Oberflächen nicht entgegenkommen. Eine leicht gebogene Tastatur, die Platz zwischen beiden Händen lässt und in welcher einzelne, oft schwerer mit den Fingern erreichbare Tasten größer dargestellt werden, würde den im Rahmen der Studie beobachteten Schreibmustern hingegen eher entsprechen.¹⁵⁵

Benutzerschnittstellendesign

Innovative Benutzerschnittstellen zwingen zu einem Umdenken nicht nur hinsichtlich der damit umsetzbaren Interaktionsmöglichkeiten, sondern auch in Bezug auf die konkrete Oberflächengestaltung. Die horizontale Ausrichtung sowie die Möglichkeit von Berührungseingaben erfordern andere Benutzerschnittstellenelemente (UI-Widgets) als das traditionelle WIMP-Paradigma. Generell sollten Tabletop-Anwendungen so einfach wie möglich aufgebaut sein und das Potential der Gesteninteraktion nutzen, um platzaufwändige (Steuer-) Elemente vermeiden und die eigentlichen Inhalte in den Fokus stellen zu können.¹⁵⁶

Bereits erläutert wurde die Relevanz der flexiblen Ausrichtung von Objekten im Hinblick auf die Nutzung von Tabletops von jeder Position um den Tisch herum. Grafische Elemente sollten möglichst von allen Seiten erkennbar sein.¹⁵⁷ Eine flexible Ausrichtung setzt auch die Möglichkeit voraus, Elemente frei auf der Oberfläche zu bewegen und zu positionieren, was insbesondere bei textuellen Inhalten von Relevanz ist. Generell wird deshalb empfohlen, beispielsweise Menüs oder andere Elemente, die von allen Benutzern bedient werden können, nicht wie bei herkömmlichen GUIs fest an einer Stelle, sondern frei auf der Oberfläche zu platzieren.¹⁵⁸

Durch das kollaborative Potential von Tabletopsystemen stellt sich darüber hinaus die Frage, ob gemeinsam genutzte Elemente lediglich einmal oder in Form von Duplikaten auf der Oberfläche zur Verfügung gestellt werden sollen. Morris und andere sehen für duplizierte Widgets in Mehrbenutzerszenarien dahingehend Potential, dass Kollision zwischen einzelnen Benutzern bei der Interaktion

155 Vgl. Findlater, Wobbrock & Wigdor (2011).

156 Vgl. Microsoft (2009, S. 32), vgl. Wigdor & Wixon (2011, S. 31).

157 Vgl. Microsoft (2009, S. 20).

158 Vgl. Apted, Collins & Kay (2009, S. 2, S. 4), vgl. Microsoft (2009, S. 19).

mit Elementen vermieden und Personalisierung ermöglicht werden kann. Die Verwendung mehrerer Instanzen von Steuerelementen nimmt hingegen auch immer Platz in Anspruch, der gegebenenfalls für die eigentliche Arbeitsaufgabe reserviert bleiben sollte. In einer vergleichenden Studie zur gemeinsamen Annotation von Photos mit *TeamTag* bevorzugten die Benutzer jedoch duplizierte Steuerelemente gegenüber zentralisierten Menüs.¹⁵⁹ Eine generelle Abwägung zwischen beiden Optionen muss selbstverständlich immer anwendungsbezogen erfolgen.

Benutzerschnittstellenelemente, mit denen durch Berührung interagiert wird, müssen darüber hinaus groß genug sein, um komfortabel mit einem Finger selektiert werden zu können (siehe auch Kapitel 3.2.2). Textuelle Informationen sollten zudem in der Regel größer als üblich dargestellt werden, um Lesbarkeit, speziell aus verschiedenen Perspektiven, zu gewährleisten. Durch diese Anforderungen ergibt sich jedoch selbst an großen Oberflächen, insbesondere in Kombination mit duplizierten Elementen, schnell das Problem der Überfrachtung der Benutzerschnittstelle (*Crowding* bzw. *Cluttering*). Strategien zur Verringerung von *Clutter* beinhalten beispielsweise die Möglichkeit, Elemente jederzeit minimieren oder entfernen zu können.¹⁶⁰ Kann eine für die Berührungsinteraktion optimierte Mindestgröße von Elementen nicht umgesetzt werden, bieten sich Alternativen wie Zoombereiche oder indirekte Selektionstechniken wie versetzte Cursors an.¹⁶¹ Letztere können jedoch, vor allem in kollaborativen Szenarien, die *Group Awareness* verringern, da durch indirekte Interaktionstechniken mit der Metapher der direkten Manipulation von Objekten gebrochen wird.¹⁶² Schließlich muss diesbezüglich, je nach Anwendungskontext, eine angemessene Balance zwischen der Lesbarkeit und Benutzbarkeit, der Übersichtlichkeit und Einfachheit sowie der Angemessenheit der Benutzerschnittstelle für kollaborative Aufgaben gefunden werden.¹⁶³

Das konkrete Design von Steuerelementen stellt dabei eine weitere Herausforderung dar. Prinzipiell sollten diese klar von den eigentlichen Inhalten unterschieden werden können.¹⁶⁴ Linear aufgebaute Menüs mit einzelnen, textuellen Einträgen, wie man sie in herkömmlichen grafischen Benutzerschnittstellen findet, erfüllen die Anforderungen, die an Tabletop-Anwendungen gestellt werden –

159 Vgl. Morris et al. (2006a).

160 Vgl. Apted, Collins & Kay (2009, S. 4).

161 Vgl. Benko & Wigdor (2010, S. 253 ff.).

162 Vgl. Shen et al. (2006, S. 40).

163 Vgl. Ryall et al. (2006a, S. 93), vgl. Apted, Collins & Kay (2009, S. 2, S. 4).

164 Vgl. Microsoft (2009, S. 32).

beispielsweise die gemeinsame Bedienung aus allen Perspektiven zu ermöglichen – nur bedingt. Morris und andere setzen deshalb in einer Version des *TeamTag*-Prototypen für gemeinsam genutzte Steuerelemente kreisförmige Menüs ein.¹⁶⁵ Eine von Graham und anderen im Rahmen von Forschungsarbeiten erstellte und 2010 veröffentlichte Widget-Bibliothek für Tabletop-Anwendungen (auf Basis des *Windows Presentation Framework* (WPF)) stellt ebenfalls ein radiales, mehrere Ebenen umfassendes Menü zur Verfügung (Abbildung 17).¹⁶⁶

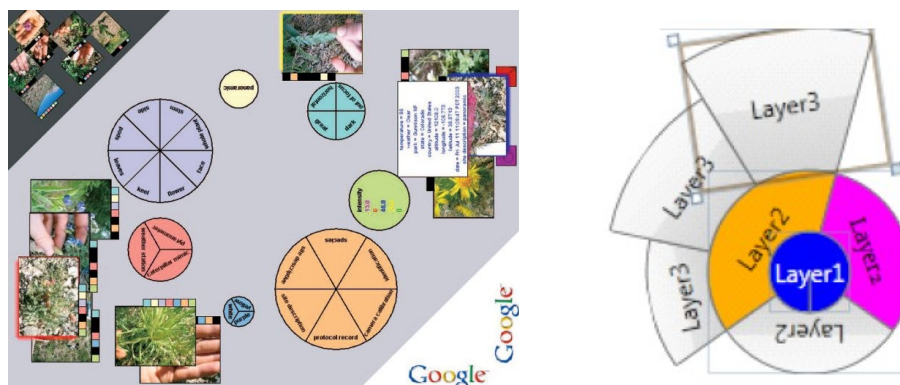


Abbildung 17: Perspektivisch unabhängige, kreisförmige Menüs in *TeamTag* (Ryall et al. 2006b, S. 58) und als Bestandteil einer *TabletopWidget*-Bibliothek (Graham et al. 2010).

Kann jeder Benutzer individuell auf personalisierte Menüs zugreifen oder kommen Kontextmenüs zum Einsatz, spielt eine perspektivenunabhängige Ausrichtung eventuell eine geringere Rolle. Dennoch stellt sich sowohl die Frage nach der optimalen Ausnutzung des auf der Oberfläche verfügbaren Platzes, insbesondere wenn das Menü mehrere Einträge oder Ebenen umfasst, als auch nach der Anpassung an die Touch-Interaktion. Hesselmann, Flöring und Schmitt schlagen mit ihren *Stacked Half-Pie Menus* eine für den Tabletop optimierte halbkreisförmige Abwandlung des klassischen hierarchischen Menüs vor (Abbildung 18).¹⁶⁷ Bei einem von Leithinger und Haller vorgestellten Prototypen hingegen werden einzelne Menüelemente auf der Basis eines vom Benutzer gezeichneten Pfades auf der Oberfläche angeordnet, um den auf dem Bildschirm oftmals knapp bemessenen Platz optimal ausnutzen zu können (Abbildung 18).¹⁶⁸ Während diese beiden Prototypen lediglich einzelne Berührungen erfordern, zeigen Kammer,

¹⁶⁵ Vgl. Morris et al. (2006a, S. 1274 f.).

¹⁶⁶ Vgl. Graham et al. (2010).

¹⁶⁷ Vgl. Hesselmann, Flöring & Schmitt (2009).

¹⁶⁸ Vgl. Leithinger & Haller (2007).

Lamack & Grohl eine Lösung, die zum Aufschalten eines ringförmigen globalen Menüs eine fünf Finger umfassende Multi-Touch-Geste erfordert (Abbildung 18).

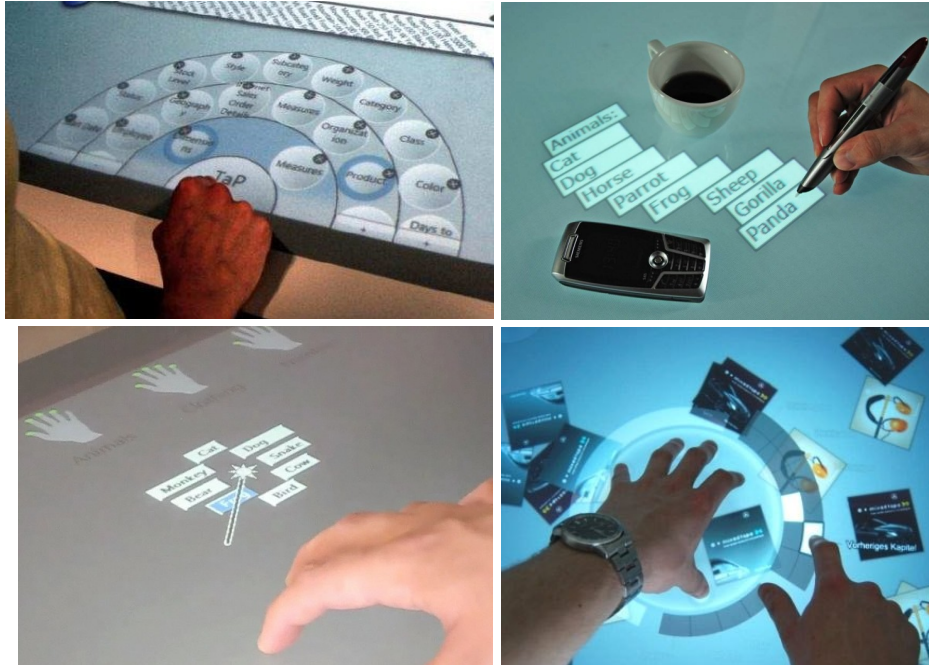


Abbildung 18: *Stacked Half-Pie-Menü* (o.l., Hesselmann, Flöring & Schmitt 2009, S. 173), *User Drawn Path-Menü* (o.r., Leithinger & Haller 2007, S. 122), *Multitouch Marker-Menü* (u.l., Lepinski, Grossmann & Fitzmaurice 2010b) und globales *Ring-Menü* (u.r., Kammer, Lamack & Grohl 2010, S. 260).

Kontextmenüs zu einzelnen Elementen können darüber hinaus über eine *Tap and Hold*-Geste aufgeschaltet werden.¹⁶⁹ Lepinski, Grossmann und Fitzmaurice zeigen schließlich im Rahmen einer Studie zu *Marking Menus*, dass der Einsatz von Multi-Touch-Gesten auch direkt bei der Menübedienung zum Einsatz kommen kann. In diesem Falle können über gerichtete Mehrfingergesten einzelne Menüpunkte in einem versetzt radial angeordneten Menü ausgewählt und durch unterschiedliche Hierarchien navigiert werden. Durch die indirekte Interaktion mit dem Menü kann an dieser Stelle zudem *Clutter* vermieden werden (Abbildung 18).¹⁷⁰ Der Einsatz insbesondere von tief verschachtelten Menüs am Tabletop ist jedoch prinzipiell in Frage zu stellen, nicht zuletzt deshalb, da ein langfristiges Ziel sein soll, dieses aus dem WIMP-Paradigma stammende Konzept durch innovative Interaktionsmethoden abzulösen.

¹⁶⁹ Vgl. Kammer, Lamack & Grohl (2010).

¹⁷⁰ Vgl. Lepinski, Grossmann & Fitzmaurice (2010a).

Von ebenso hoher Relevanz für das Design von Tabletopbenutzerschnittstellen ist auch das Feedback, das durch die Interaktion mit Widgets vom System gegeben wird. Grundsätzlich sollte darauf geachtet werden, dass Änderungen des Systemzustandes für alle Benutzer bemerkbar sind, um die *Awareness* und *Feed-through* zu fördern (siehe auch Kapitel 2.4 und 3.2.2). Bei der Auswahl von Elementen durch Berührung sollte ersichtlich sein, dass ein bestimmtes Element selektiert wurde, beispielsweise durch eine Veränderung der Form, Beschaffenheit oder Farbe des Objektes.¹⁷¹ Bei der Ausführung von Manipulationsgesten sollten Veränderungen von Elementen bezüglich deren Form, Größe oder Ausrichtung letztlich ebenso unmittelbar sichtbar sein.¹⁷²

Tangible Interaction

Ein großer Teil der Forschungsarbeiten zur Tabletoptechnologie entspringt dem Bereich der *Tangible Interaction*, welche sich mit der physikalischen Darstellung digitaler Informationen und der Interaktion mit diesen tangiblen Elementen (*Tangibles*) auseinandersetzt. Die horizontale Oberfläche von Tabletops bietet ein vielversprechendes Einsatzgebiet für *Tangibles* im Kontext diverser Anwendungen. Elemente zur Kontrolle und Manipulation von Bildelementen sind hier ebenso denkbar, wie die Erkennung und/oder Augmentierung von am Tisch verwendeten Objekten, wie Papier oder Mobiltelefonen. Zentrales Argument für den Einsatz von *Tangibles* auf dem Tabletop ist deren inhärente Physikalität und Greifbarkeit, die, im Gegensatz zu in der Regel zweidimensional dargestellten virtuellen Objekten, eine „natürlichere“ Interaktion im Raum ermöglichen soll, da sie neben haptischem Feedback ein reiches Manipulationsvokabular bereitstellen, für das auf grundlegende mentale Modelle und Fähigkeiten aus dem Alltag zurückgegriffen werden kann.¹⁷³ Für kollaborative Szenarien wird davon ausgegangen, dass der Einsatz von *Tangibles* die *Awareness* (siehe auch Kapitel 2.4) innerhalb der Gruppe fördern kann, da die Manipulation von physikalischen Objekten zur Sichtbarkeit der von den einzelnen Gruppenmitgliedern am Tisch ausgeführten Handlungen beiträgt.¹⁷⁴

Bereits zahlreiche der frühen Tabletopsysteme, wie der *DigitalDesk*, der *ActiveDesk* oder der *metaDESK* (siehe Kapitel 3.1), machen sich diese Eigenschaften von *Tangibles* zu Nutze. Der bekannteste Tabletop, der *Microsoft Surface*, kann ebenfalls mit

¹⁷¹ Vgl. Microsoft (2009, S. 19, S. 22).

¹⁷² Vgl. Microsoft (2009, S. 30 ff.).

¹⁷³ Vgl. Terrenghi et al. (2008, S. 138 f.).

¹⁷⁴ Vgl. Scott, Grant & Mandryk (2003, S. 11 f.).

Byte-Tags versehene Objekte erfassen. Diese und weitere Forschungsprototypen, wie der inzwischen auch kommerziell erhältliche *reactTable* zeigen, wie *Tangibles* sinnvoll zur Manipulation von digitalen Informationen an Tabletopsystemen eingesetzt werden können. Der 2007 entwickelte *reactTable*, besteht aus einem Tabletopsystem mit runder Oberfläche, mit der mit Hilfe von physikalischen Tokens – auch kollaborativ – interagiert werden kann. Diese *Tangibles* fungieren als Steuerelemente für einen virtuellen Synthesizer – durch das Berühren und Drehen der greifbaren Objekte sowie durch die Veränderung der räumliche Anordnung der *Tangibles* untereinander können vielfältige Klangstrukturen erzeugt werden (Abbildung 19).¹⁷⁵

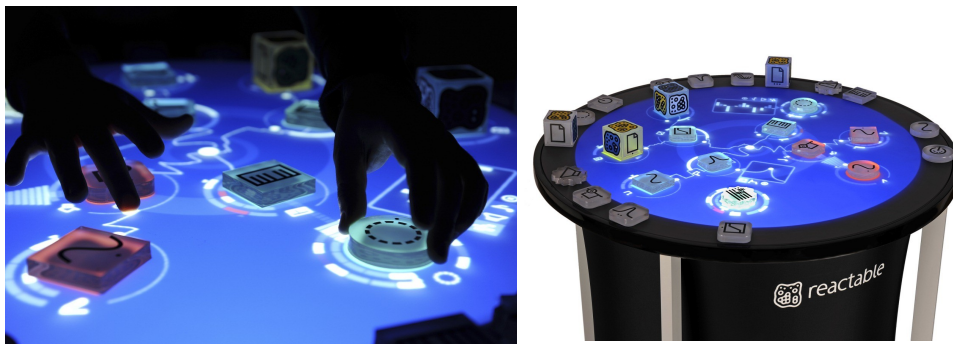


Abbildung 19: Tangible Interaction am Tabletop am Beispiel des *reactTable* (Boldrin 2011)

Weitere Anwendungsszenarien sind beispielsweise die Umsetzung von Brettspielen am Tabletop mit Hilfe von *Tangibles* (*Tangible Tabletop Gaming*), die physikalische Umsetzung von Steuer- und Anzeigeelementen für Anwendungen am Tabletop, wie zum Beispiel Foto- oder Informationsbrowser, das Anzeigen von zusätzlichen digitalen Informationen zu bestimmten physikalischen Tokens sowie das Augmentieren von Dokumenten auf Tabletops um zusätzliche digitale Inhalte (*Hybrid Documents*).¹⁷⁶ Neben passiven Objekten, die vom Benutzer bewegt werden müssen, werden ebenfalls Prototypen mit ‚aktiven *Tangibles*‘ entwickelt, die durch Einsatz von Robotik in der Lage sind, sich automatisch auf der Tabletopoberfläche zu bewegen.¹⁷⁷

Neben diesen im Allgemeinen für die Entwicklung von Benutzerschnittstellen an Tabletops relevanten Problem- und Forschungsfeldern stellt insbesondere das

¹⁷⁵ Vgl. Jordà et al. (2007).

¹⁷⁶ Vgl. Idziorek (2007), vgl. Hilliges, Baur & Butz (2007), vgl. Terrenghi et al. (2008), vgl. Shaer et al. (2010), vgl. Nagel et al. (2009), vgl. Deininghaus et al. (2010), vgl. Steimle et al. (2010).

¹⁷⁷ Vgl. Inami et al. (2010).

kollaborative Arbeiten an horizontalen, interaktiven Oberflächen Designer und Entwickler vor neue Herausforderungen. Dabei relevante Aspekte seien in der Folge kurz vorgestellt.

3.2.4 Kollaborative Nutzung von Tabletops

Eines der zentralen Konzepte der Forschungsarbeiten rund um die Tabletoptechnologie ist die Möglichkeit des kollaborativen Arbeitens. Der Einsatz von großen, horizontalen, berührungsempfindlichen Oberflächen mit Multi-Touch-Funktionalität erlaubt gleichzeitiges und gleichberechtigtes Arbeiten mit digitalen Inhalten. Im Rahmen der Konzeption und Umsetzung von Anwendungen, die im Hinblick auf das gemeinsame Interagieren mehrerer Personen entwickelt werden, sollten einige Bereiche, welche im Folgenden näher beleuchtet werden, besondere Beachtung erfahren.

Persönliche Bereiche

Um den Prozess des kollaborativen Arbeitens an Tabletopsystemen besser verstehen und unterstützen zu können, lohnt ein Blick auf das Verhalten von Gruppenteilnehmern beim gemeinsamen Arbeiten an traditionellen, nicht-interaktiven Tischen. Neben der kommunikativen Funktion der manuellen Ausrichtung von Objekten (siehe auch Kapitel 3.2.3) zeigen Beobachtungen derartiger Szenarien insbesondere die Relevanz von „Territorialität“ (*territoriality*) für Koordination und Kollaboration durch die Aufteilung des Arbeitsbereiches in persönliche und geteilte Bereiche.¹⁷⁸ Scott, Carpendale und Inkpen konnten diese Beobachtungen im Rahmen einer 2004 veröffentlichten Studie zur Kollaboration an traditionellen Tischen bestätigen: bei der gemeinsamen Arbeit bilden die Teilnehmer ohne explizite Absprache unterschiedliche Bereiche auf dem Tisch, um ihre Interaktion mit für die Aufgabe relevanten Objekten und den anderen Teilnehmern zu organisieren. Scott und andere unterscheiden dabei persönliche Bereiche (*personal areas*), Gruppenbereiche (*group areas*) und Bereiche zur (temporären) Ablage von Objekten (*storage areas*) (Abbildung 20).¹⁷⁹ Der Bereich unmittelbar vor einem Teilnehmer bildet dabei in der Regel den persönlichen Bereich, welcher üblicherweise nur von der jeweiligen Person für individuelle Aufgaben, die unabhängig von der Gruppe ausgeführt werden können, sowie Lese- und Schreibaufgaben, verwendet wird. Soziale Normen hindern andere Teilnehmer in der Regel daran, mit

¹⁷⁸ Vgl. Tang (1991), vgl. Kruger & Carpendale (2002).

¹⁷⁹ Vgl. Scott, Carpendale & Inkpen (2004, S. 296).

persönlichen Bereichen anderer Benutzer zu interagieren. Dennoch können die persönlichen Bereiche anderer Teilnehmer dazu dienen, deren Handlungen zu beobachten und zukünftige Aktionen im Sinne der *Group Awareness* zu antizipieren. Der zentrale Bereich des Tisches sowie Areale zwischen einzelnen, individuell genutzten Bereichen werden hingegen zur Bearbeitung der eigentlichen Gruppenaufgabe genutzt. In den untersuchten Szenarien (u.A. Lösen eines Puzzles, Erstellung eines Raumplans) wurden zudem angrenzend an und inmitten von individuell und gemeinsam genutzten Arealen mobile Bereiche identifiziert, die der temporären Ablage von einzelnen Elementen dienen. Die Grenzen der einzelnen Bereiche sind dabei fließend und werden neben der Anzahl der Gruppenteilnehmer am Tisch und dessen Größe letztlich vor allem durch die darin befindlichen Objekte und die damit stattfindenden Interaktionshandlungen bestimmt.¹⁸⁰

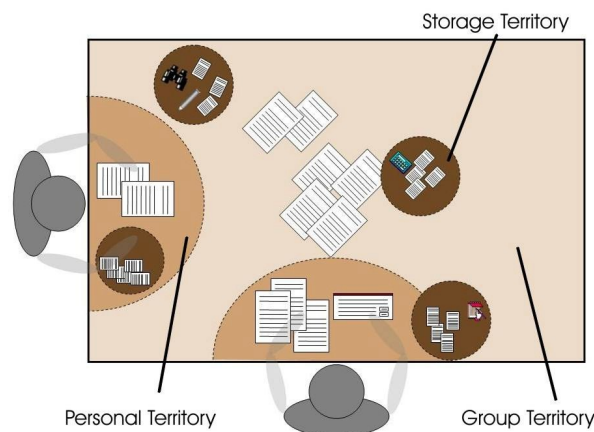


Abbildung 20: Aufteilung in individuell, gemeinsam und zur Ablage genutzte Arbeitsbereiche beim kollaborativen Arbeiten an einem traditionellen Tisch (Habelski 2004, S. 24).

Für das Design digitaler Tabletops folgern Scott, Carpendale und Inkpen, dass Benutzer in ihrer Intuition, eine Oberfläche im Rahmen von kollaborativen Aufgaben in diese einzelnen Bereiche zu separieren, unterstützt werden müssen. Das heißt, interaktive Oberflächen müssen groß genug sein, um sowohl Individual- als auch Gruppenarbeit zu ermöglichen und das Ablegen von gerade nicht benötigten Elementen zu erlauben. Um das individuelle Arbeiten zu unterstützen, sollten Objekte einfach in persönliche Bereiche transportiert werden können und Werkzeuge für individuell durchgeführte Aufgaben leicht erreichbar sein.¹⁸¹ Da

¹⁸⁰ Vgl. Scott, Carpendale & Inkpen (2004, S. 296 f.).

¹⁸¹ Vgl. Scott, Carpendale & Inkpen (2004, S. 302 ff.).

die Grenzen der einzelnen Bereiche fließend sind und, je nach Position eines Benutzers am Tisch sowie Aufgabenfortschritt, wechseln können, wird eine feste Aufteilung der Tabletopoberfläche in individuell und gemeinsam genutzte Bereiche nicht empfohlen, da dies die Benutzer in einer speziell dem Aufgabenkontext angepassten Aufteilung der Oberfläche behindert.¹⁸²

Privatsphäre und Sicherheit

Neben der Möglichkeit, individuelle Arbeitsbereiche an Tabletops zu bilden, erfordert ein kollaboratives Szenario, je nach Arbeitsaufgabe, auch die Beschäftigung mit Themen wie Privatsphäre (*privacy*) und Sicherheit (*security*). Das Arbeiten in persönlichen Bereichen kann bereits eine gewisse Form von Privatsphäre bereitstellen, jedoch können auch am Tabletop Situationen auftreten, in denen Benutzer ihre Handlungen vor anderen Benutzern verbergen wollen oder müssen. Je nach Arbeitsaufgabe kann es nötig sein, Gruppenteilnehmern die Möglichkeit zu geben, zusätzlich private Räume zu schaffen sowie private Ein- und Ausgabekanäle zur Verfügung zu stellen. Letzteres ist insbesondere für Szenarien interessant, welche den Austausch sensibler Informationen zur Identifikation und Authentifizierung von Benutzern erfordern.

Im Rahmen des Prototyps *RoomPlanner* zur Raumplanung werden private Bereiche beispielsweise durch das angewinkelte Platzieren der Hand auf der Tischoberfläche in Form einer Geste geschaffen, wodurch persönliche Informationen in das vor anderen Personen verdeckten Areal des Tisches sowie auf die angewinkelte Hand des Benutzers projiziert werden können (Abbildung 21).¹⁸³ Eine ähnliche Methode zeigen Kim und andere mit *ShieldPIN*: dort lässt erst die Ausführung einer entsprechenden Handgeste auf dem Tabletop ein Feld zur Eingabe einer PIN (*Personal Identification Number*) erscheinen (Abbildung 21).¹⁸⁴ Eine weitere Möglichkeit ist die Anzeige von verschiedenen Informationen für unterschiedliche Perspektiven am Tabletop. Dies geschieht bei einem von Sakurai und anderen umgesetzten Prototypen durch die Kombination von zwei rückseitigen Projektoren mit einem drehbaren Polarisationsfilter (*revolving polarizer*) als Teil des Projektionsaufbaus sowie zusätzlichen Polarisationsfiltern in von Benutzern getragenen Brillen. Durch diesen Systemaufbau ist es möglich, bestimmte Informationen jeweils für einzelne Benutzer (un-) sichtbar zu machen.¹⁸⁵ Die Eingabe sensibler In-

182 Vgl. Scott & Carpendale (2010, S. 382).

183 Vgl. Wu & Balakrishnan (2003, S. 198 f.).

184 Vgl. Kim et al. (2010, S. 1096).

185 Vgl. Sakurai et al. (2009).

formationen, beispielsweise zur Authentifizierung einzelner Benutzer, stellt eine weitere Herausforderung dar. Die Eingabe textueller Informationen am Tabletop bietet hierfür je nach Kontext eventuell nicht das vom Benutzer gewünschte Ausmaß an Sicherheit. An dieser Stelle bietet sich die Möglichkeit der Erkennung personalisierter Gesten als Ersatz für textuelle Passwörter an.¹⁸⁶ Ebenso denkbar ist die Identifizierung einzelner Benutzer über biometrische Daten. Dies kann beispielsweise über Erfassung der individuellen Form der Hand des Benutzers, welche im Prototyp *HandsDown* von Schmidt, Chong und Gellersen zur Identifizierung einzelner Personen am Tabletop herangezogen wird, geschehen.¹⁸⁷

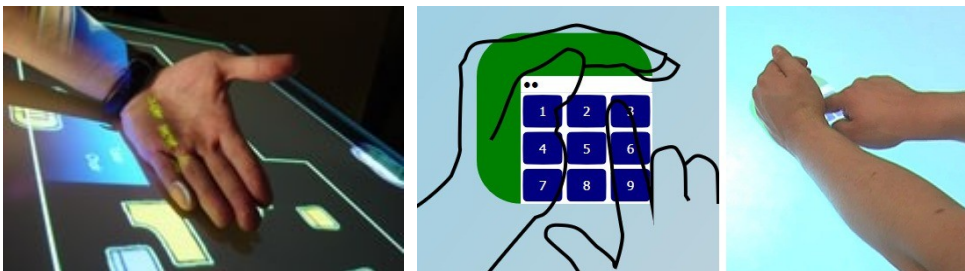


Abbildung 21: Handgesten zur Schaffung von privaten Räumen zur Aus- (Wu & Balakrishnan 2003, S. 199) und Eingabe (Kim et al. 2010, S. 1096) sensibler Informationen am Tabletop.

Darüber hinaus stellt sich die Frage, wie nach einer erfolgreichen Identifizierung die Interaktion einzelner Benutzer am Tabletop aussehen kann. Die Wahrnehmung unterschiedlicher Benutzerrollen und die Anzeige personalisierter Informationen setzen in der Regel voraus, dass der Tisch die Interaktionshandlungen nach Benutzer unterscheiden kann, was jedoch, wie in der Folge noch erläutert wird, bisher kaum ein Tabletopsystem unterstützt. Eine derartige Unterscheidung ermöglicht der ebenfalls von Schmidt, Chong und Gellersen vorgestellte Prototyp *IdLenses* insofern, als für jeden durch seine Handform identifizierten Benutzer ein personalisiertes Linsen-Widget aufgeschaltet wird, das, je nach Kontext, personalisierte Informationen anzeigen sowie individuelle Funktionen bereitstellen kann. Als ein Anwendungsfall im Kontext Sicherheit ist hier beispielsweise die automatische Authentifizierung bei Webportalen durch Positionieren des personalisierten Widgets auf den Login-Bereich einer Webseite genannt, wodurch die direkte Eingabe sensibler Daten umgangen werden kann.¹⁸⁸

¹⁸⁶ Vgl. Schöning, Rohs & Krüger (2008, S. 2).

¹⁸⁷ Vgl. Schmidt, Chong & Gellersen (2010a).

¹⁸⁸ Vgl. Schmidt, Chong & Gellersen (2010b).

Eine weitere Möglichkeit zur Erhöhung von Privatsphäre und Sicherheit stellt die Ergänzung des Tabletops um externe Geräte wie PDAs, Laptops oder Smartphones dar.¹⁸⁹ Diese können einerseits private Räume schaffen, in denen Benutzer ungestört individuellen Arbeitsaufgaben nachgehen können, sowie andererseits einen zusätzlichen Kanal für den Austausch von Informationen bereitstellen.¹⁹⁰ Eine Vielzahl von Projekten und Forschungsarbeiten beschäftigt sich im Kontext von *Single Display Groupware* (SDG) und *Multi Display Environments* (MDE) mit der Erweiterung von öffentlichen Displays um private Ein- und Ausgabegeräte, wie in Kapitel 4 beschrieben werden wird.

Unterscheidung von Benutzern

Ein in den meisten derzeit auf dem Markt befindlichen Tabletopsystemen noch nicht umgesetztes Konzept ist die Unterscheidung der (Berührungs-) Interaktionshandlungen nach einzelnen Benutzern. Während der *DiamondTouch* durch seinen kapazitiven Touchscreen und entsprechende Empfänger in den Stühlen der Benutzer die Unterscheidung von bis zu vier Personen am Tisch ermöglicht (siehe Kapitel 3.1), stellt die Differenzierung von Benutzern an Tabletopsystemen, die wie der *Microsoft Surface* oder der *Evoluce ONE* mit optischen Erkennungsverfahren arbeiten, bisher noch eine relativ große Herausforderung dar.

Die fehlende Unterscheidung nach dem Ursprung einer Berührung schränkt Anwendungen für Tabletops dahingehend ein, dass beispielsweise unterschiedliche Interaktionsmodi nicht unterstützt werden können. Ein Beispiel hierfür sind Zeichenprogramme, bei denen zuerst eine bestimmte Funktion ausgewählt wird, welche in der Folge die Aktionen, die mit dem Cursor ausgeführt werden, bestimmt. Ohne die Unterscheidung einzelner Benutzer würde ein selektierter Modus (sofern nicht auf einen bestimmten Bereich beschränkt) an einem Tabletop immer auch für alle anderen Benutzer gelten müssen.¹⁹¹ Ist keine Unterscheidung nach Benutzer möglich, sollte der Einsatz von unterschiedlichen Interaktionsmodi am Tabletop vermieden werden.¹⁹²

Ein weiteres Problem einer fehlenden Benutzerdifferenzierung ist die Identifizierung einzelner Benutzer in Szenarien, in denen personalisierte Informationen von Relevanz sind. Die von Ryall und anderen für den *DiamondTouch* umgesetzten *iDwidgets* machen sich beispielsweise die Möglichkeit der Unterscheidung von

¹⁸⁹ Vgl. Schöning, Rohs & Krüger (2008, S. 2).

¹⁹⁰ Vgl. Shoemaker & Inkpen (2001).

¹⁹¹ Vgl. Scott & Carpendale (2006, S. 26).

¹⁹² Vgl. Wigdor & Wixon (2011, S. 41).

Personen am Tabletop dahingehend zu Nutzen, dass Steuerelemente wie Menüs oder Buttons je nach Benutzer, der mit ihnen interagiert, individuell angepasst werden und somit unterschiedliche Aktionen auslösen und persönliche Informationen anzeigen können.¹⁹³ Tabletopsysteme, die die Unterscheidung von Benutzern nicht nativ unterstützen, müssen an dieser Stelle auf andere Methoden zurückgreifen. Eine Lösung kann hier das Tragen von spezieller mit Infrarot-LEDs ausgestatteter Hardware an der Hand oder am Arm sein, welche für jeden Benutzer ein individuelles Bitmuster sendet, das von den Kameras im Tabletop zusätzlich zu den Berührungspunkten erkannt wird. Prototypen wie der *IR Ring* oder *IdWristbands* setzen diese Technik bereits ein, um Interaktionen am Tabletop nach Benutzer unterscheiden zu können (Abbildung 22).¹⁹⁴



Abbildung 22: Technologien zur Unterscheidung und Identifizierung von Benutzern am Tabletop: *idWristband* (Meyer & Schmidt 2010, S. 277) und *TouchID*-Handschuhe (Marquardt et al. 2011, S. 9).

Eine weitere Möglichkeit ist das Erfassen von einzelnen Benutzern über Bytecodes, welche üblicherweise zur Erkennung von *Tangibles* verwendet werden. Das von Marquardt und anderen entwickelte System *TouchID* setzt dafür mit mehreren Bytecodes versehene Handschuhe ein, die es ermöglichen, bei der Interaktion an einem *Microsoft Surface* Berührungen und Gesten einzelne Finger und Hände sowie schließlich auch Benutzer zu unterscheiden (Abbildung 22).¹⁹⁵

Globale Aktionen, Dialoge und Systemmeldungen

Bei der gleichzeitigen Interaktion mehrerer Personen mit einem Tabletopsystem können bestimmte Aktionen, die eine Wirkung auf die gesamte Anwendung oder mehrere Benutzer haben, problematisch sein. Ein einzelner Benutzer sollte bei-

¹⁹³ Vgl. Ryall et al. (2006b, S. 58 ff.).

¹⁹⁴ Vgl. Roth, Schmidt & Güldenring (2010), vgl. Meyer & Schmidt (2010).

¹⁹⁵ Vgl. Marquardt et al. (2011).

spielsweise nicht in der Lage sein, eine Ansicht, die alle Benutzer teilen, zu verändern, sei es durch Hineinzoomen in einen Bereich oder durch Verschieben der Ansicht, da dadurch der Arbeitsprozess der anderen Benutzer gestört werden kann.¹⁹⁶ Bei einer von Shen und anderen für den Tabletop entwickelten Anwendung zur Exploration von räumlich-geographischen Daten können Benutzer zum Beispiel über eine individuelle Auswahl in einzelne Bereiche hineinzoomen und diese verschieben, ohne Einfluss auf die allgemeine Ansicht zu nehmen.¹⁹⁷ Weitere kritische globale Aktionen, die alle Benutzer betreffen, wie das Schließen einer Anwendung oder das Entfernen von mehreren Objekten, sollten die Zustimmung aller Teilnehmer erfordern. Dies kann beispielsweise durch kollaborative / kooperative Gesten erfolgen, die von mehreren Benutzern gleichzeitig ausgeführt werden müssen.¹⁹⁸ Eine weitere Möglichkeit ist der Einsatz von Voting-Widgets, welche die Zustimmung aller Benutzer bezüglich der Ausführung einer globalen Aktion erfassen.¹⁹⁹

Während globale Aktionen und Systemmeldungen, die alle Benutzer betreffen, so gestaltet werden sollten, dass sie die Aufmerksamkeit aller Benutzer erfordern, sollten von einzelnen Benutzern angestoßene Aktionen und dabei erfolgende Systemrückmeldungen den Arbeitsprozess der anderen Personen am Tisch so wenig wie nötig stören.²⁰⁰ Dialoge und Systemmeldungen, die sich auf eine bestimmte Interaktionshandlung eines Benutzers beziehen, sollten demnach, wenn möglich, auch nur für diesen Benutzer angezeigt werden, beispielsweise als Einblendung an der Stelle der letzten bzw. maßgeblichen Interaktion dieses Benutzers.

Somit lässt sich festhalten, dass es noch eine Vielzahl von unbeantworteten Fragen bezüglich der (kollaborativen) Interaktion an digitalen Tabletops sowie der konkreten Gestaltung und Umsetzung von Anwendungen für derartige Systeme gibt. Dabei sollte neben dem konkreten Anwendungskontext schließlich auch immer ein besonderes Augenmerk auf die bisher für horizontale interaktive Oberflächen identifizierten Problemfelder gelegt werden und Aspekte wie die flexible Ausrichtung von Bildelementen, das Design von adäquaten Gesten oder die Unterstützung der Benutzer bezüglich flexibler Segmentierung der Arbeitsoberfläche in die Konzeption mit einfließen. So sind einige der hier identifizierten Problem- und Forschungsfelder selbstverständlich auch für die im Rahmen die-

196 Vgl. Wigdor & Wixon (2011, S. 35).

197 Vgl. Shen et al. (2006, S. 41 f.).

198 Vgl. Morris et al. (2006b).

199 Vgl. Morris et al. (2004, S. 264).

200 Vgl. Apted, Collins & Kay (2009, S. 2).

ser Arbeit umgesetzten Anwendung zum Brainstorming und Mind-Mapping am Tabletop relevant und finden sich im Anwendungskonzept wieder, dessen Entwicklung in Kapitel 5 näher beleuchtet wird. Zunächst soll jedoch noch ein Blick auf die bisher im Kontext von Tabletop-Technologie entwickelten Anwendungen zum (kollaborativen) Brainstorming und Mind-Mapping geworfen werden.

3.3 Brainstorming und Mind-Mapping am Tabletop

Das kollaborative Bearbeiten von Aufgaben ist wohl einer der vielversprechendsten Anwendungskontexte für Tabletopsysteme. Neben dem grundsätzlichen Einsatz von interaktiven Tischen im Meetingkontext (beispielsweise als Teil von *Roomware* und kollaborativen Umgebungen, siehe dazu Kapitel 3.1) finden sich in den letzten Jahren auch einige prototypische Anwendungen speziell zur (kollaborativen) Durchführung von Brainstorming Sitzungen sowie von Mind-Mapping und verwandten Verfahren wie Konzept-Mapping an interaktiven Oberflächen.²⁰¹ Eine Auswahl dieser Projekte soll in der Folge kurz vorgestellt werden, um anhand der dabei umgesetzten Designlösungen Erweiterungsmöglichkeiten aufzuzeigen, die letztlich die Motivation für die im Rahmen der Arbeit umgesetzte prototypische Anwendung liefern.

Als eines der ersten Projekte mit Fokus auf Unterstützung von Problemlösungsprozessen in Gruppen an interaktiven Oberflächen ist das 2002 im Kontext der *i-LAND* Umgebung (siehe Kapitel 3.1) entstandene *BEACH*-Projekt zu nennen. An einem vertikalen, berührungsempfindlichen Display können dort Benutzer im Rahmen der Anwendung *BeachMap* Ideen durch handschriftliche Eingabe mit einem digitalen Stift generieren und in Form von Mind-Maps strukturieren. Die Bearbeitung der Mind-Map kann in diesem Fall bereits durch mehrere Benutzer gleichzeitig über Stiftgesten erfolgen (Abbildung 23).²⁰²

Das 2006 im Rahmen einer Projektarbeit an der Ludwig-Maximilians-Universität München (LMU) von David Kim entwickelte System *BrainStorm* ergänzt darüber hinaus vertikale Displays um einen horizontalen Tabletop, der, im Gegensatz zum *BEACH*-System, als primäres Eingabemedium dient, während das digitale Whiteboard zur nachträglichen Strukturierung und der Visualisierung der während des Brainstorming bzw. *Brainwriting* gesammelten Ideen verwendet wird (Abbildung 23). In einer ersten Phase können Benutzer am Tabletop durch Interaktion mit einem digitalen Stift Notizen in Form von digitalen Post-Its erstellen und bearbeiten, um diese Ideen schließlich in einer zweiten Phase am Whiteboard zu organisieren und zu sortieren. Während am Tabletop Post-Its erstellt, verschoben, geworfen, editiert, kopiert und gelöscht werden können, zei-

²⁰¹ Vgl. Prante, Magerkurth & Streit (2002), vgl. Baraldi, Bimbo & Valli (2006), vgl. Kim (2006), vgl. Altorfer (2007), vgl. Buisine et al. (2007), vgl. Wang, Ghanam & Maurer (2008), vgl. Hunter & Maes (2008), vgl. Donker & Meixner (2009), vgl. Oppl & Stary (2009), vgl. Da Luz et al. (2010), vgl. Forster (2010, S. 111, S. 115), vgl. Cowie (2010), vgl. Geyer et al. (2011).

²⁰² Vgl. Prante, Magerkurth & Streit (2002).

gen die mit dem Tabletop verbundenen vertikalen Displays Kopien aller Post-Its an und ermöglichen das Erstellen von Clustern aus mehreren Post-Its sowie deren Verbindung durch Linien zur Strukturierung des Ideenraums.²⁰³ Im Rahmen einer vergleichenden Benutzerstudie, bei der das System mit einer papierbasierten *Brainwriting*-Technik verglichen wurde, zeigte sich kein signifikanter Unterschied in Bezug auf die Quantität der Ideen oder die Anzahl und Länge von während des *Brainwritings* entstandenen Assoziationsketten. Den subjektiven Bewertungen der Benutzer nach zu urteilen wurde hingegen das elektronische *Brainwriting* bevorzugt und die damit erstellten Ideen als qualitativ hochwertiger bewertet. Die verwendeten Funktionen wurden überwiegend als „einfach zu erlernen“ erachtet.²⁰⁴ Ein entscheidender Nachteil des *BrainStorm* Systems ist jedoch, dass es nur zwei Berührungspunkte gleichzeitig erkennen kann, weshalb die Interaktion auf Single-Touch-Gesten und zwei Personen an fixen Positionen am Tabletop beschränkt bleiben muss. Die Orientierung der Post-Its ist dabei abhängig von der Position des Benutzer und kann nicht verändert werden.²⁰⁵



Abbildung 23: *BeachMap* an der *DynaWall* (Prante, Magerkurth & Streitz 2002, S. 112) und *BrainStorm* an Tabletop und Whiteboard (Hilliges et al. 2007, S. 140).

Der 2007 an der Universität Freiburg (Schweiz) für den *DiamondTouch* entwickelte Prototyp *TableMind* ermöglicht darüber hinaus die gleichzeitige Interaktion von vier Benutzern mit einer Mind-Mapping Anwendung. Die Erstellung von Ideen erfolgt, ähnlich wie bei *BrainStorm*, durch das direkte Beschreiben bzw. Bemalen von digitalen Post-Its mit dem Finger, die Interaktion mit den einzelnen Ideen erfolgt gleichermaßen durch Single-Touch-Gesten. Durch das Übereinanderschieben von zwei Post-Its, die nach ihrer Erstellung in die Mitte des Tabletops

²⁰³ Vgl. Kim (2006, S. 13 ff.).

²⁰⁴ Vgl. Hilliges et al. (2007, S. 192 ff.).

²⁰⁵ Vgl. Kim (2006, S. 13).

bewegt wurden, können Verbindungen zwischen einzelnen Ideen erstellt und somit eine Mind-Map aufgebaut werden. Die Orientierung der einzelnen Post-Its ist hier ebenfalls auf eine bestimmte Perspektive festgelegt, jedoch lässt sich die Ansicht der gesamten Mind-Map drehen. Darüber hinaus besitzt jeder der vier Benutzer einen privaten Bereich, in dem bereits beschriebene Post-Its temporär abgelegt werden können. *TableMind* unterstützt somit alle drei der von Scott und Carpendale identifizierten Arbeitsbereiche (siehe Kapitel 3.2.4), wenn auch nicht mit dem geforderten Maß an Flexibilität, da die einzelnen Bereiche fest definiert sind (Abbildung 24).²⁰⁶ Im Rahmen einer Benutzerstudie, bei der *TableMind* unter anderem mit papierbasiertem Mind-Mapping verglichen wurde, konnte bestätigt werden, dass eine Anwendung wie *TableMind* laut der subjektiven Bewertungen der Benutzer intuitiv und einfach zu benutzen ist und das kollaborative Mind-Mapping positiv unterstützen kann.²⁰⁷

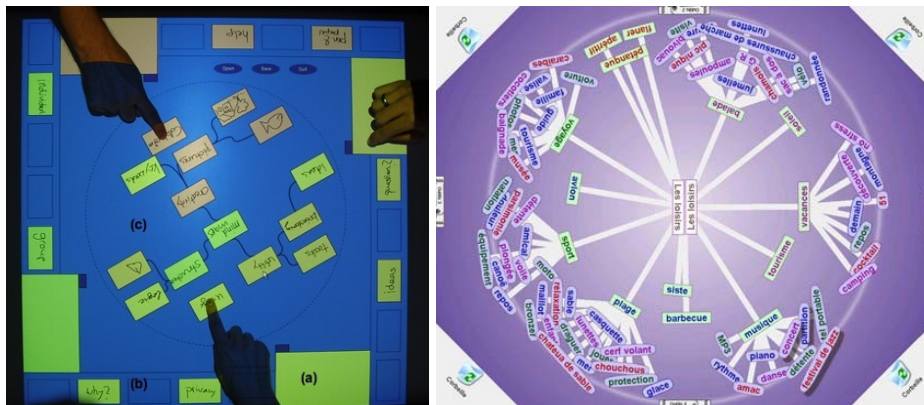


Abbildung 24: Kollaboratives Mind-Mapping am *DiamondTouch* bei *TableMind* (Altorfer 2007, S. 18) und *Tabletop Mind Maps* (Montferrat 2007, S. 1).

In Bezug auf die Benutzerzufriedenheit können Buisine und andere für eine 2007 ebenfalls für den *DiamondTouch* entwickelte Anwendung namens *Tabletop Mind Maps* ähnlich positive Ergebnisse im Vergleich zum papierbasierten Mind-Mapping vorweisen. Zudem wurde eine erhöhte Kollaboration am Tabletop in Form von kommunikativen Gesten beobachtet. Bezüglich der Quantität und Qualität von Ideen konnte zwischen elektronischem und papierbasiertem Mind-Mapping, wie bereits auch schon bei der Evaluation von *TableMind*, kein signifikanter Unterschied festgestellt werden.²⁰⁸ Dies lässt sich möglicherweise auch auf das konkrete Design der Anwendung zurückführen. Die in der Mitte des Bildschirms po-

²⁰⁶ Vgl. Altorfer (2007).

²⁰⁷ Vgl. Altorfer (2007, S. 34 ff.).

²⁰⁸ Vgl. Buisine et al. (2007, S. 28 ff.).

sitionierte Mind-Map, deren einzelne Ideen automatisch radial ausgerichtet werden, kann durch das doppelte Antippen einer Idee erweitert werden, jedoch erfolgt die textuelle Eingabe dieser Ideen nicht direkt am Tabletop, sondern durch nur einen Benutzer über eine externe Tastatur, was letztlich der Idee, durch elektronische Unterstützung Produktionsblockaden zu verhindern, zuwider läuft (siehe Kapitel 2.3). Die Orientierung der Ideen ist darüber hinaus durch deren radiale Anordnung festgelegt, die zentrale Idee der Mind-Map wird jedoch immerhin zur besseren Lesbarkeit aus zwei Perspektiven dargestellt (Abbildung 24).

Der 2008 am *Massachusetts Institute Of Technology* (MIT) entwickelte Tabletop-prototyp *WordPlay* zur elektronischen Unterstützung von Ideengenerierung und Entscheidungsfindung in Gruppen erlaubt schließlich auch die Interaktion über Multi-Touch-Gesten sowie die flexible Positionierung und Ausrichtung von Ideen auf der Tabletopoberfläche. Das Hinzufügen von Ideen erfolgt über Spracheingabe sowie alternativ über eine virtuelle Tastatur. Einzelne Ideen können bewegt, gelöscht, rotiert und skaliert werden (Abbildung 25). Die Verbindung einzelner Textbausteine in Form von Mind-Maps wird nicht unterstützt, jedoch können zu einzelnen Begriffen mit Hilfe einer Wisch-Geste durch Anbindung von *WordNet* und *ConceptNet3* Assoziationen generiert werden.²⁰⁹ Eine 2009 an der Technischen Universität Dresden entwickelte Anwendung zur „Unterstützung von Kreativitätssitzungen“²¹⁰ erlaubt ebenfalls das freie Platziere und Ausrichten von Ideen in Form von digitalen Karteikärtchen, die über eine virtuelle Tastatur beschrieben werden und unterschiedlich farbig markiert werden können. Darüber hinaus können Verbindungen zwischen einzelnen Ideen erstellt werden (Abbildung 25). Die Interaktion mit dem Tabletop erfolgt hier jedoch nicht über Berührungsgesten, sondern lediglich über digitale Stifte; zudem ist die Benutzeroberfläche für zwei Personen konzipiert, wodurch Steuerelemente wie bei *TableMind* und *Tabletop Mind Maps* eine fixe Position am Tisch einnehmen.²¹¹

Ein 2011 vorgestellter Prototyp namens *FireStorm*, der an der Universität Sydney entwickelt wurde, versucht das allgemeine Problem der langsamen Texteingabe an Tabletops durch den Einsatz von kabellosen Tastaturen zu lösen. Jeder der vier Benutzer am Tisch kann somit zu jeder Zeit Ideen über eine am Rand des Tabletops platzierte Tastatur eingeben. Einzelne Ideen, die farblich nach Benutzer unterschieden werden, können durch eine Berührungsgeste in einen Container verwandelt werden, der nach der eigentlichen Brainstormingphase zum

209 Vgl. Hunter & Maes (2008).

210 Donker & Meixner (2009, S. 83).

211 Vgl. Donker & Meixner (2009), vgl. Meixner (2008).

Gruppieren weiterer, thematisch zugehöriger Ideen verwendet werden soll. Auch bei *FireStorm* sind einzelne Ideen frei positionierbar und rotierbar; die Ausrichtung der Textfelder erfolgt jedoch nach deren Erstellung zunächst automatisch in einer kreisförmigen Anordnung um den Mittelpunkt herum, um die *Group Awareness* zu fördern. Eine Evaluation des Systems konnte zeigen, dass sowohl die Anzahl der am Tabletop generierten Ideen als auch die Anzahl der generierten Oberkategorien in der Sortierungsphase ebenso hoch bzw. höher als die Anzahl der Ideen war, die in einem Szenario an einem traditionellen Whiteboard generiert wurden.²¹²

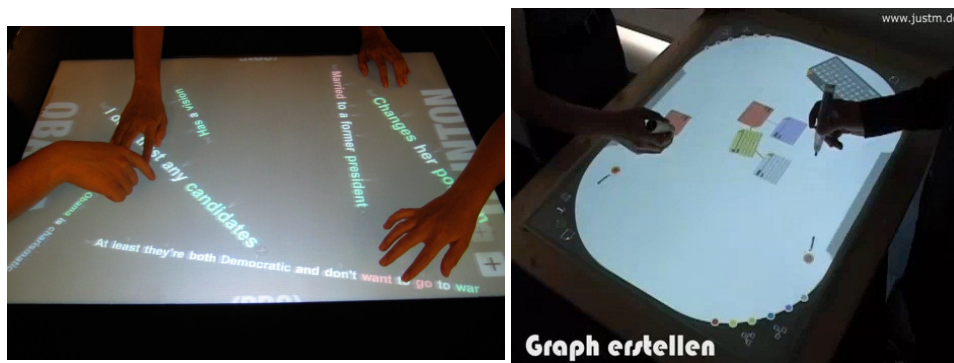


Abbildung 25: Flexible Positionierung und Ausrichtung von Ideen bei *WordPlay* (Hunter & Maes 2008, S. 2) und am *cgTable* (Meixner 2008).

Während all diese Prototypen lediglich die Erstellung textueller Ideen sowie gegebenenfalls von Skizzen erlauben, ermöglichen Prototypen für das Brainstorming und Mind-Mapping am Tabletop wie *MindFlow* (2009) oder der *AffinityTable* (2011) darüber hinaus auch die Einbindung von weiteren Medien wie Videos oder digitalen Bildern.²¹³ Am *AffinityTable* erfolgt die Erfassung von Ideen mit einer hybriden Methodik: Während der Ideengenerierungsphase werden wie bei einem traditionellen *Brainwriting* Notizzettel aus Papier beschriftet. Da dies mit digitalen Stiften geschieht, wird durch das nachfolgende Auflegen eines Post-Its auf die Tabletopoberfläche eine digitale Kopie der Inhalte des Notizzettels erzeugt. Eine duplizierte Ansicht des gesamten Arbeitsbereichs wird dabei an einem zusätzlichen vertikalen Display dargestellt, um eine weitere Visualisierung der Ideen sowie einen Bereich, in dem einzelne Teile der Arbeitsfläche vergrößert dargestellt werden können, zur Verfügung zu stellen. Hierfür unterstützt das System die Interaktion mit *Tangibles*, die zum Hineinzoomen in einen bestimmten Teil der Ar-

²¹² Vgl. Clayphan et al. (2011).

²¹³ Vgl. Da Luz et al. (2010, S. 243), vgl. Geyer et al. (2011).

beitsfläche verwendet werden können (Abbildung 26). Weitere Anwendung finden mit Bytecodes getaggte, physikalische Objekte bei der Selektion und Gruppierung von einzelnen Ideen oder beim Löschen von digitalen Notizzetteln. Zusätzlich unterstützt das System schlagwortbasiertes Bildretrieval, welches durch das Auflegen von Papier-Post-Its auf einen bestimmten Bereich des Tabletops angestoßen wird und somit das Einbinden von digitalen Medien zusätzlich zu den digitalisierten Post-Its erlaubt.²¹⁴



Abbildung 26: Ideengenerierung und -strukturierung mit hybriden Medien und Tangibles am *AffinityTable* (Geyer et al. 2011, S. 480).

Zusammenfassend lässt sich somit feststellen, dass in den letzten Jahren bereits zahlreiche prototypische Anwendungen zum Brainstorming und Mind-Mapping am Tabletop umgesetzt worden sind, die jeweils unterschiedliche Lösungen für allgemein an horizontalen Oberflächen auftretende Probleme, wie die Eingabe von Texten oder die Unterstützung der Interaktion von allen Seiten, vorschlagen und neben der Erfassung, Manipulation und Strukturierung von Ideen zusätzliche Funktionalität zur Unterstützung von Gruppenarbeit bereitstellen können. Sofern für die einzelnen Systeme Benutzerstudien durchgeführt wurden, zeigt sich in der Regel, dass sich bezüglich der Quantität und Qualität von Ideen sowie der subjektiven Benutzerzufriedenheit ähnliche Ergebnisse erzielen lassen wie in traditionellen Brainstorming-/Brainwriting- und Mind-Mapping-Szenarien, was im Hinblick auf die zusätzlichen Vorteile des digitalen Mediums durchaus positiv bewertet werden kann.

Während bei der Mehrzahl der vorgestellten Systeme eine Art Notizzetteltapher zum Einsatz kommt und Aspekte wie die freie Positionierung und Ausrichtung von Bildelementen sowie die Unterstützung von Multi-Touch-

²¹⁴ Vgl. Geyer et al. (2011).

Gesten in der Regel unterstützt werden, unterscheiden sie sich doch häufig in den Bereichen Texteingabe sowie der Segmentierung in Areale unterschiedlicher Funktion (beispielsweise Bereiche zur Individual- und Gruppenarbeit) und der damit verbundenen Unterstützung unterschiedlicher Arbeitsprozesse innerhalb kollaborativer Szenarien. So arbeiten Systeme wie *Tabletop Mind Maps* oder *FireStorm* mit zusätzlichen Tastaturen, um die Eingabe von Texten in gewohnter Art zur unterstützen, während andere Prototypen wie *TableMind*, der *AffinityTable* oder *BrainStorm* die handschriftliche Eingabe von Text erlauben. Wieder andere setzen auf virtuelle Tastaturen oder gar wie das System *WordPlay* auf Spracheingabe. Ebenso heterogen zeigt sich die Unterstützung von privaten Räumen und Individualarbeit. Systeme wie *Brainstorm* oder *TableMind* stellen feste Bereiche für Individual- und Gruppenarbeit zur Verfügung, während Prototypen wie *WordPlay* oder der *cgTable* die von Scott und Carpendale geforderte Flexibilität bezüglich der Segmentierung der Benutzeroberfläche in derartige Bereiche unterstützen. Am *AffinityTable* oder bei *FireStorm* sind die persönlichen Bereiche der Benutzer darüber hinaus größtenteils außerhalb des Tabletops lokalisiert.

An dieser Stelle soll die vorliegende Arbeit ansetzen. Die Ergänzung um zusätzliche Eingabegeräte, die neben der Eingabe von Texten auch weitere Funktionen bereitstellen können, kommt bisher in keinem der prototypischen Anwendungen zum Einsatz. Die Erweiterung von horizontalen Oberflächen um private Eingabegeräte, beispielsweise in Form von Smartphones, ist darüber hinaus nicht nur für den problematischen Bereich der Texteingabe interessant, sondern birgt ebenso Potential zur Unterstützung von Individualarbeit und privaten Räumen, wie in der Folge näher erläutert wird.

4 Multi-Device-Interaktion mit Tabletop und Smartphones

Die im Rahmen der vorliegenden Arbeit umgesetzte prototypische Anwendung zum Brainstorming und Mind-Mapping am Tabletop *Multi/Touch/Device MindMapper* ergänzt die horizontale, kollaborativ genutzte Oberfläche des interaktiven Tisches um externe Geräte in Form von Smartphones, welche zur Eingabe von neuen Ideen und der Unterstützung von Individualarbeit dienen. Zunächst wird einer allgemeinen Begriffsdefinition folgend anhand der Vorstellung ähnlicher Lösungen aus dem Bereich der *Single Display Groupware* (SDG), Multi-Device-Interaktion (MDI) und kollaborativen Umgebungen versucht werden, den *Multi/Touch/Device MindMapper* in einen größeren Forschungskontext einzubetten, um schließlich auf die für die Motivation zu einer Kombination von Tabletops und Smartphones in diesem Aufgabenkontext relevanten Aspekte eingehen zu können.

4.1 Single Display Groupware, Multi-Display-Umgebungen und Multi-Device-Interaktion

Versucht man, eine passende Überschrift für den Bereich von Anwendungen, die die Kombination und Kommunikation unterschiedlicher digitaler Geräte miteinander erfordern oder ermöglichen, zu finden, stößt man schnell auf eine Vielzahl von Begriffen, die bisher zur Klassifikation derartiger Systeme verwendet wurden. Deshalb soll zunächst versucht werden, diese Begrifflichkeiten zu klären, um schließlich auf konkrete Multi-Device-Anwendungen in kollaborativen Szenarien einzugehen.

4.1.1 Definitionen und Abgrenzung

Stewart, Bederson und Druin sprechen bei der Kombination von großen, gemeinsam genutzten Displays mit externen Eingabegeräten für jeden Benutzer von *Single Display Groupware* (SDG).²¹⁵ Als „single display“ wird dabei ein gemeinsam genutzter Computer mit einem entsprechenden Ausgabebildschirm (beispielsweise ein Desktoprechner oder auch ein digitales Whiteboard) bezeichnet. Dieses Display kann von allen Benutzern gleichzeitig bedient werden, beispielsweise mit Hilfe mehrerer Mäuse, Tastaturen oder auch durch damit verbundene externe

²¹⁵ Vgl. Stewart, Bederson & Druin (1999).

Geräte wie PDAs. Diese externen Geräte sind jedoch in der Regel nur als voneinander unabhängige Eingabekanäle gedacht, die Ausgabe von zentralen Informationen erfolgt üblicherweise für alle Benutzer am gemeinsam genutzten Display. Eine kollaborativ genutzte Anwendung für einen digitalen Tabletop ließe sich demnach prinzipiell auch bereits ohne weitere Eingabekanäle als SDG verstehen, da es sich hier um ein gemeinsam genutztes Display mit mehreren parallelen Eingabekanälen handelt. SDG wurde als Alternative zu verteilter *Groupware* vorgeschlagen, da sie im Gegensatz zu *Groupware*, die verteilt auf mehreren einzelnen Rechnern läuft, die Kollaboration dadurch fördern soll, dass alle Gruppenteilnehmer an einem gemeinsamen Bildschirm arbeiten und direkt miteinander kommunizieren können.²¹⁶

Spätestens mit der Verbreitung von *Roomware* und der Integration mehrerer unabhängiger digitaler Geräte in kollaborative Umgebungen (siehe Kapitel 3.1) taucht neben *Single Display Groupware* ein weiterer Begriff auf: *Multi-Display Environments* (MDEs). Während bei SDG der Fokus auf einem gemeinsam genutzten Rechner bzw. Bildschirm liegt, der als alleiniges Ausgabemedium dient, steht bei MDEs die Einbettung mehrerer digitaler Geräte (und deren Displays) in eine spezielle (kollaborative) Umgebung im Fokus. Die Ausgabe von relevanten Informationen muss nicht mehr nur auf dem zentralen Display erfolgen, da die externen Eingabegeräte (wie PDAs, Laptops, Tablet-PCs) in der Regel über einen eigenen Bildschirm verfügen und so ebenfalls als Ausgabedisplays genutzt werden können. Die Kombination von privaten, mobilen Geräten mit großen, gemeinsam genutzten Displays ist hier dennoch häufig anzutreffen. An dieser Stelle ist jedoch weniger die individuelle Kontrolle eines Cursors auf einem gemeinsam genutzten Bildschirm, sondern vielmehr die Kommunikation zwischen mehreren Displays und der damit stattfindende (möglichst bidirektionale) Transfer von digitalen Informationen von zentraler Bedeutung.²¹⁷

Eine allgemeinere Bezeichnung derartiger Systeme, welche den Aspekt der Integration in eine bestehende Umgebung außer Acht lässt, ist *Coupled Displays* (dt. gekoppelte Anzeigen).²¹⁸ Terrenghi, Quigley und Dix klassifizieren einzelne Systeme an dieser Stelle unter anderem anhand einer Skala, die von losen bis hin zu eng gekoppelten Displays reicht. Während unter „eng gekoppelten Displays“ hier solche Systeme zu verstehen sind, für welche die Verbindung zwischen den einzelnen Anzeigen fest und damit nicht frei konfigurierbar ist (wie Videoleinwände

216 Vgl. Stewart, Bederson & Druin (1999, S. 286,).

217 Vgl. Nacenta et al. (2009, S. 172 f.).

218 Vgl. Terrenghi, Quigley & Dix (2010, S. 584).

aus mehreren Bildschirmen oder eine Ansammlung von fest an bestimmten Plätzen installierten Desktoprechnern), werden Systeme, die die Existenz und Verbindung zwischen einzelnen Displays nicht voraussetzen, sondern je nach Verfügbarkeit und Notwendigkeit einsetzen, als lose gekoppelt bezeichnet.²¹⁹

Im Fall von lose gekoppelten Anzeigen und von MDEs lässt sich schließlich auch erstmals von wirklicher Multi-Device-Interaktion sprechen, da die einzelnen Geräte in der Regel voneinander unabhängig als eigenständige Systeme betrachtet werden können, und nicht nur als zusätzlicher Eingabekanal für ein gemeinsam genutztes Display fungieren. An dieser Stelle wird der im Rahmen dieser Arbeit verwendete Begriff der Multi-Device-Interaktion wie folgt verstanden: Multi-Device-Interaktion bezeichnet die Möglichkeit oder Notwendigkeit der Interaktion von Benutzern mit mehreren, in der Regel eigenständigen, miteinander verbundenen Geräten im Kontext einer spezifischen Anwendung.²²⁰ Die Kombination von Tabletop und Smartphones im Rahmen der Anwendung *Multi/Touch/Device MindMapper* kann somit ebenfalls als ein Beispiel für Multi-Device-Interaktion verstanden werden. In der Folge seien einige bisherige Prototypen vorgestellt, welche diese Form der Interaktion gleichermaßen versuchen umsetzen.

4.1.2 Multi-Device-Anwendungen in kollaborativen Szenarien

Eine frühe prototypische Anwendung, die das Prinzip der Multi-Device-Interaktion in einem kollaborativen Szenario bereits in Teilen umsetzt, ist das 1998 von Myers, Stiel und Gargiuio vorgestellte System *Pebbles (PDAs for the Entry of Both Bytes and Locations from External Sources)*. Der Bildschirm eines Desktoprechners dient als gemeinsam genutztes Display, während einzelne Benutzer jeweils über einen PDA mit dem Rechner interagieren können. Die zentrale Anwendung des *Pebbles*-Systems ist der *Remote Commander*, welcher die mit dem PDA stattfindende Interaktion auf Maus- und Tastatureingaben des Desktoprechners abbildet. Ein hierfür umgesetzter Anwendungsfall erlaubt beispielsweise das Navigieren in und Editieren von digitalen Präsentationsfolien über den PDA durch mehrere Benutzer. Eine weitere Anwendung ist *PebblesDraw* – ein Grafikeditor, bei dem jeder Benut-

²¹⁹ Vgl. Terrenghi, Quigley & Dix (2010, S. 584 f.).

²²⁰ Hiervon abzugrenzen ist jene Definition von Multi-Device Interaktion, welche Aspekte der Interaktion, die beim Interaktionsdesign von Anwendungen für unterschiedliche Hardwareplattformen (Desktoprechner, Mobiltelefon, PDA, Tablet-PC etc.) beachtet werden müssen, bezeichnet (oftmals im Kontext von *Multi-Device Interfaces/Systems* bzw. *Divisible Interfaces* verwendet).

Vgl. Hutchings & Pierce (2006), vgl. Richter et al. (2006).

zer über einen eigenen Cursor verfügt, der über den PDA gesteuert werden kann. Die Ausgabe von Informationen auf dem PDA ist nicht vorgesehen.²²¹

Ein 1998 von Rekimoto vorgestellter Prototyp verbindet ein digitales Whiteboard mit mehreren PDAs. Im Gegensatz zu *Pebbles* wird hier nicht nur ein Cursor kontrolliert, sondern der PDA stellt weitere Funktionalität zur Verfügung. Da die Texteingabe am vertikal ausgerichteten Whiteboard als umständlich empfunden wird, ermöglicht der Prototyp die Eingabe von Texten an den Palmtops. Diese Textobjekte können ebenso wie andere digitale Objekte mit Hilfe der *Pick-and-Drop*-Technik auf das Whiteboard übertragen werden. Hierbei selektiert der Nutzer mit dem Stylus ein Objekt am PDA und transferiert es durch Selektion des Zielbereichs am vertikalen Display auf das Whiteboard. Als ein Anwendungsfall wird hier auch das Brainstorming in Gruppen genannt.²²²

Die Kombination von Laptops mit einem gemeinsam genutzten Display kann darüber hinaus einen Zugewinn an Funktionalität auf Seite des Eingabekanals bieten. Der Systemprototyp *Note&Share*, 2010 von Streng und anderen vorgestellt, verbindet beispielsweise Laptops und ein vertikales Display zur gemeinsamen Problemlösung. Benutzer können zunächst auf ihren persönlichen Rechnern, die als private Räume fungieren, Vorschläge sammeln und diese schließlich im Rahmen des Meetings auf dem vertikalen Display zur öffentlichen Diskussion stellen und dort weiter ordnen und strukturieren. Der Transfer von Ideen auf das digitale Whiteboard erfolgt an dieser Stelle mit einer *Drag and Drop*-Geste auf einen bestimmten Bereich der persönlichen Arbeitsfläche am Laptop.²²³ Während sich hier der Austausch auf Textobjekte beschränkt, erlaubt das 2003 von Everitt vorgestellte Tabletopsystem *UbiTable* den Austausch von diversen digitalen Objekten wie Dokumenten oder digitalen Bildern mit dem Tisch gekoppelten Laptops. Benutzer verbinden sich über ihren Laptop mit dem Tabletop und können sich über ein frei positionierbares Icon am interaktiven Tisch einen persönlichen Bereich reservieren, in dem die vom Laptop transferierten Objekte angezeigt werden. Der Austausch erfolgt hier, ebenso wie bei *Note&Share*, über das Positionieren von Objekten in einen bestimmten Bereich der Anwendung am Laptop.²²⁴

Roomwareumgebungen wie *i-LAND* oder der *NiCE Discussion Room* verbinden schließlich mehrere unterschiedliche digitale Geräte miteinander.²²⁵ So können

221 Vgl. Myers, Stiel & Gargiuio (1998).

222 Vgl. Rekimoto (1998).

223 Vgl. Streng et al. (2010).

224 Vgl. Everitt (2003).

225 Vgl. Prante, Streitz & Tandler (2004), vgl. Haller et al. (2010).

bei *i-LAND* im Rahmen der unterschiedlichen Anwendungen der dafür umgesetzten Software-Infrastruktur *BEACH* (*Basic Environment for Active Collaboration with Hypermedia*) digitale Objekte zwischen einem vertikalen Display (*DynaWall*), einem Tabletop (*InteracTable*) und in Stühle (*CommChairs*) sowie flexible Tische (*ConnecTables*) integrierte Rechnern ausgetauscht werden.²²⁶ Eine weitere für *i-LAND* umgesetzte Anwendung namens *PalmBeach* erlaubt darüber hinaus die Übertragung von zuvor auf einem PDA erstellten textuellen Ideen auf das vertikale Display.²²⁷ Der Transfer der Daten vom Palmtop zur *DynaWall* wird dabei über das Betätigen eines speziellen Buttons, während der PDA auf das vertikale Display gerichtet wird, angestoßen. Der Austausch digitaler Objekte zwischen den übrigen Roomwarekomponenten erfolgt über die unmittelbare Synchronisation eines mit allen verbundenen Komponenten geteilten Anzeigebereichs, in den digitale Objekte, die geteilt werden sollen, aus dem persönlichen Arbeitsbereich gezogen werden können.²²⁸

Ein Blick auf die bisher im Kontext von Multi-Device-Interaktion umgesetzten Anwendungen zur Unterstützung von Meetings und kreativen Prozessen zeigt somit, dass es bereits einige Ansätze zur Verbindung von großen, gemeinsam genutzten Displays mit mobilen Geräten gibt. Ein Großteil dieser Systeme fungiert als *Proof-of-concept*-Prototypen; für einige der Anwendungen, wie beispielsweise *Note&Share* oder die Roomwarekomponenten in *i-LAND*, sind jedoch auch vergleichende Studien durchgeführt worden. So konnten Streititz, Rexroth und Holmer für eine frühe Version der *i-LAND*-Komponenten, das System *DOLPHIN*, bestehend aus einem digitalen Whiteboard und in einen Tisch integrierte Arbeitsstationen, eine signifikant höhere Anzahl an qualitativ hochwertigeren Ideen im Rahmen eines Problemlösungsprozesses im Vergleich zum gleichen Aufgabenszenario, lediglich unterstützt durch das digitale Whiteboard oder die verbundenen Arbeitsstationen, feststellen.²²⁹ Streng und andere haben *Note&Share* im Rahmen einer Studie zur moderierten Problemlösung in der Gruppe im Vergleich zu einem herkömmlichen, nicht-interaktiven Whiteboard sowie zu einem papierbasierten Szenario evaluiert und dabei feststellen können, dass die Mehrheit der Benutzer *Note&Share* als für eine effektive Problemlösung am hilfreichsten (65 Prozent) sowie als das von ihnen persönlich präferierte System (84 Prozent), jedoch nicht als das am einfachsten benutzbare System (36 Prozent im Vergleich zu

226 Vgl. Prante, Streititz & Tandler (2004, S. 48)

227 Vgl. Prante, Magerkurth & Streititz (2002, S. 112 f.).

228 Vgl. Prante, Streititz & Tandler (2004, S. 50 f.).

229 Vgl. Streititz, Rexroth & Holmer (1997).

Whiteboard 48 Prozent) bewerten.²³⁰ Das Potential von Multi-Device-Interaktion für die Arbeit in Gruppensituationen ist somit ebenso wenig von der Hand zu weisen wie der Bedarf an weiterer Forschung zu geeigneten Anwendungen und Benutzerschnittstellen in diesem Bereich.

Während einige der vorgestellten Systeme mobile Geräte mit gemeinsam genutzten Displays im Kontext von Meetingszenarios verbinden, ist die Kombination von Smartphones mit Tabletops speziell im Kontext von Brainstorming- und Mind-Mapping Szenarien bisher noch nicht untersucht worden. Warum gerade diese Kombination im Rahmen von Anwendungen zur elektronischen Unterstützung von Kreativitätstechniken in Gruppen geeignet sein kann, sei in der Folge erläutert.

²³⁰ Vgl. Streng et al. (2010, S. 791 f.).

4.2 Motivation für die Kombination von Tabletop und Smartphones

Es gibt zahlreiche Argumente für den Einsatz von Tabletops in kollaborativen Szenarien allgemein sowie speziell zur elektronischen Unterstützung von Brainstorming und Mind-Mapping: die horizontale Fläche unterstützt eine gleichberechtigte Interaktion durch alle Gruppenteilnehmer und kann somit unter anderem dazu beitragen, Produktionsblockaden zu verhindern. Darüber hinaus fördert die Möglichkeit der direkten Manipulation von Bildelementen die *Awareness* innerhalb der Gruppe und die *Face-to-Face*-Kollaboration (siehe auch Kapitel 2.4).

Dennoch zeigen Tabletopsysteme auch gewisse Schwachstellen – so ist die Eingabe von Text am Tabletop häufig problematisch. Externe Hardware wie Tastaturen benötigen Platz und können durch das Brechen mit der Metapher der direkten Manipulation den Arbeitsprozess stören; handschriftliche Eingaben sind langsam und erfordern zusätzliche Erkennungssoftware. Die Bedienung von virtuellen Tastaturen ist ebenfalls zeitintensiv und umständlich, was unter anderem auch auf den Mangel an von traditionellen Tastaturen gewohntem haptischen Feedback zurückzuführen ist (siehe Kapitel 3.2.3).

Ein weiterer Aspekt ist die Unterstützung sowohl von Individual- (*taskwork*) als auch von Gruppenarbeit (*teamwork*). Beide Prozesse umfassen Aktivitäten, die für ein erfolgreiches kollaboratives Arbeiten Voraussetzung sind – so wird *taskwork* als die Art von Arbeit definiert, die zur eigentlichen Lösung der Aufgabe durch Einzelne benötigt wird, wie beispielsweise das Generieren von Ideen. *Teamwork* umfasst hingegen solche Aktivitäten, die benötigt werden, um als Gruppe arbeiten zu können, wie Koordination, Kommunikation und das Aufrechterhalten der *Awareness* innerhalb der Gruppe.²³¹ Die große, horizontale Oberfläche eines Tabletop bietet das Potential zur Unterstützung von Gruppenarbeit. Individualarbeit ist jedoch je nach Größe der Oberfläche und Anzahl der Benutzer gegebenenfalls nur eingeschränkt möglich, da womöglich Platz auf der Oberfläche rar ist oder es zu Kollisionen zwischen einzelnen Benutzern mit unterschiedlichen Aufgaben kommt. Beispielsweise können in der Regel anwendungsferne Aufgaben – wie eine zusätzliche Informationssuche im *World Wide Web* während eines Meetings – nicht gelöst werden, ohne den Arbeitsprozess der anderen Gruppenmitglieder zu stören – wie zum Beispiel durch das Öffnen eines separaten Browserfensters. Die

²³¹ Vgl. Wallace et al. (2009, S. 569 f.).

räumlich auf den Bereich des Tabletop beschränkte Zusammenarbeit, bei der alle Aktionen eines Benutzers von den anderen Gruppenmitgliedern beobachtet werden können, kann darüber hinaus auch aufgrund von sozialen Effekten negativen Einfluss auf die Kollaboration nehmen, wie bereits zuvor ausgeführt wurde (siehe Kapitel 3.2.1).

An dieser Stelle zeigt sich das Potential externer Geräte, wie privat genutzter Smartphones in Kombination mit interaktiven, öffentlichen Oberflächen. Als Smartphone wird ein

Mobiltelefon mit erweitertem Funktionsumfang [bezeichnet]. Dazu zählen neben der Telefonie und Short Message Service (SMS) üblicherweise Zusatzdienste wie Electronic Mail (E-Mail), World Wide Web (WWW), Terminkalender, Navigation sowie Aufnahme und Wiedergabe audiovisueller Inhalte. Auf Smartphones laufen gegenüber herkömmlichen Mobiltelefonen komplexere Betriebssysteme wie etwa Symbian OS, Blackberry OS oder das iPhone OS. Die hierdurch geschaffene Möglichkeit zur Installation weiterer Applikationen durch den Endnutzer verleiht Smartphones einen erweiterbaren und individualisierbaren Funktionsumfang.²³²

Obwohl die meisten aktuell auf dem Markt befindlichen Smartphones zur Eingabe von Texten lediglich virtuelle Tastaturen bereitstellen, deren Bedienung ebenfalls problematisch sein kann, kann die Texteingabe über ein Smartphone statt direkt am Tabletop dennoch in Bezug auf unterschiedliche Aspekte von Vorteil sein. Zum einen kann angeführt werden, dass die Verbreitung von Smartphones in den letzten Jahren, nicht zuletzt durch die Popularität des *Apple iPhone*, stark zugenommen hat und viele Personen als Benutzer von Mobiltelefonen mit Touchscreens bereits Erfahrungen hinsichtlich der Texteingabe an solchen Geräten gesammelt haben dürften, was von Tabletops nicht unbedingt behauptet werden kann. McAdam und Brewster argumentieren an dieser Stelle zusätzlich, dass mobile Geräte im Gegensatz zu Tabletops mehr Möglichkeiten zur Verbesserung und persönlicher Anpassung der Texteingabe bieten. Neben den klassischen Hardwaretastaturen von Mobiltelefonen gibt es inzwischen je nach Präferenz eine Vielzahl unterschiedlicher Texteingabemethoden an mobilen Geräten mit Touchscreens, darunter zahlreiche Varianten der klassischen virtuellen Bildschirmastatur und unterschiedliche Interaktionstechniken zur Selektion von einzelnen Buchstaben, sowie alternative Texteingabemöglichkeiten wie gesten- und symbolbasierte Texteingabe, Handschriftenerkennung oder Spracheingabe-

232 Sjurts (2011).

be.²³³ Darüber hinaus ist die Umsetzung von zusätzlichen taktilen und akustischen Rückmeldungen bei der Texteingabe an mobilen Geräten – beispielsweise in Form von Vibrationen und Tastaturgeräuschen – einfacher und mit weniger Auswirkungen auf die Arbeit anderer Gruppenteilnehmer umzusetzen als an einem von mehreren Personen genutzten Tabletop.²³⁴ Ein weiteres Argument für die Texteingabe über mobile Geräte am Tabletop ist die Möglichkeit, dadurch *Clutter* zu verhindern (siehe Kapitel 3.2.3). Wenn mehrere Personen an einem Tabletop gleichzeitig Text über virtuelle Tastaturen eingeben wollen, kann es schnell zu Platzproblemen kommen. Die Verlagerung der Texteingabe auf externe Geräte kann somit zur Reduzierung von Benutzerschnittstellenelementen genutzt werden, wodurch eventuell auch mehr Benutzer als üblich an einem Tabletop arbeiten könnten.²³⁵ Neben der Möglichkeit der Texteingabe kann dieser zusätzliche Eingabekanal selbstverständlich auch für den Transfer weiterer für den Problemlösungsprozess relevanter Daten (wie Dokumente, digitale Bilder oder Videos) auf den Tabletop genutzt werden. Chehimi und Ruzkio zeigen dies anhand eines Prototyps, der den Transfer von digitalen Fotos von Smartphones auf einen Tabletop über eine einfache *Flick*-Geste erlaubt.²³⁶

Neben den Möglichkeiten, die sich bezüglich der Eingabe von Text am Tabletop mit Smartphones sowie für den mobilen Datenaustausch ergeben, können externe Geräte in einem Brainstorming- bzw. Mind-Mapping-Szenario die Individualarbeit unterstützen.²³⁷ Man stelle sich vor, dass während der Phase der Ideengenerierung zusätzliche Informationen benötigt werden, die lediglich über das Internet zugänglich sind. Sofern dieser Beschaffungsprozess nicht in die eigentliche Tabletop-Anwendung integriert ist, hat das Erwerben dieser Information über den Tabletop in der Regel disruptiven Charakter, da eventuell die primäre Anwendung minimiert oder geschlossen werden muss, um an die Informationen zu gelangen. Ein externes Gerät mit Internetzugang ermöglicht hingegen parallele Informationsbeschaffung, ohne die Arbeit der restlichen Gruppenteilnehmer zu beeinflussen.²³⁸ Ein weiterer Aspekt ist die Möglichkeit, einzelne Aufgaben auch außerhalb der eigentlichen kollaborativen Arbeitszeit auf dem externen Gerät auszuführen und die dabei erzielten Ergebnisse zum Zeitpunkt des Zusammentreffens in der Gruppe mit anderen Teilnehmern zu tei-

233 Vgl. Dunlop & Masters (2009).

234 Vgl. McAdam & Brewster (2009, S. 509).

235 Vgl. McAdam & Brewster (2009, S. 510).

236 Vgl. Chehimi & Ruzkio (2010).

237 Vgl. Rekimoto (1998, S. 346).

238 Vgl. Rekimoto (1998, S. 345).

len. Hier wäre es vorstellbar, bereits vor einer Brainstormingsitzung Ideen mit Hilfe eines externen Gerätes festzuhalten und diese schließlich im Rahmen der kollaborativen Arbeit auf ein gemeinsam genutztes Display zum Zwecke der Diskussion und Organisation zu transferieren.²³⁹

An dieser Stelle wird durch das externe Gerät zudem ein privater Raum geschaffen, in dem der Benutzer gegebenenfalls weniger gehemmt interagieren kann. Fehler oder Zögern einzelner Personen sind an großen Displays wie Tabletops für alle anderen Benutzer sichtbar, was wiederum negative Effekte, wie die Angst vor Bewertung, fördern kann (siehe Kapitel 2.2). Ein privates Eingabegerät ermöglicht hier einen persönlichen Bereich, in dem der Benutzer ungestört interagieren kann, und welcher gegebenenfalls sogar eine Form von gewünschter Anonymität im Aufgabenkontext schaffen kann. Dies ermöglicht zusätzlich das unbeobachtete Navigieren in privaten Daten sowie einen sicheren Eingabekanal, beispielsweise für sensible Authentifizierungsinformationen.²⁴⁰ Ist der Aspekt der Anonymität von Eingaben über externe Geräte jedoch weniger relevant für einen bestimmten Aufgabenkontext, kann ein derartiges Multi-Device-Szenario auch die Unterscheidung bestimmter Eingaben nach Benutzer erleichtern (siehe Kapitel 3.2.4).

Der Einsatz von externen, mobilen Geräten fördert darüber hinaus die Mobilität der Benutzer am Tabletop und in der Arbeitsumgebung, da sie nicht mehr gezwungen sind, lediglich am Tabletop selbst und womöglich gar an nur einer bestimmten Stelle zu arbeiten. Schließlich sind auch auf den ersten Blick triviale Aspekte, die sich aus der (gemeinsamen) Interaktion mit berührungsempfindlichen Displays ergeben, anzuführen. Ryall und andere sowie Kray und andere berichten im Rahmen ihrer Untersuchungen zur Interaktion an Tabletops mit und ohne externe Geräte von Benutzern, die bei der mit öffentlichen Displays durch mehrere Personen stattfindenden Berührungsinteraktion Bedenken bezüglich der Hygiene solcher Systeme haben und demnach die Interaktion über indirekte Eingabemethoden, wie ein eigenes Eingabegerät, bevorzugen würden.²⁴¹

Die Erweiterung von Tabletops um externe Eingabegeräte wie Smartphones kann jedoch auch negative Effekte auf die Kollaboration haben. In erster Linie besteht an dieser Stelle die Gefahr, dass sich die Konzentration der einzelnen Benutzer zu sehr auf ihr persönliches Eingabegerät richtet und die kollaborative Arbeit am Tabletop dadurch vernachlässigt wird. Des Weiteren kann die

239 Vgl. Greenberg, Boyle & Laberge (1999, S. 55 f.), vgl. Prante, Magerkurth & Streitz (2002, S. 112).

240 Vgl. Wallace & Scott (2008, S.59), vgl. Finke et al. (2010, S. 19).

241 Vgl. Ryall et al. (2006a, S. 92 f.), vgl. Kray et al. (2010, S. 246).

Verlagerung der Aufmerksamkeit auf sekundäre Displays die *Awareness* innerhalb der Gruppe verringern, da Gruppenteilnehmer durch Fokussierung auf die Individualarbeit am Smartphone die Handlungen anderer Benutzer am Tabletop nicht mehr direkt wahrnehmen, und die am externen Gerät ausgeführten Interaktionshandlungen generell vor den anderen Gruppenteilnehmern verborgen bleiben.²⁴² Letztlich bleibt auch zu klären, ob die Eingabe von Texten über externe Geräte in der Tat schneller und komfortabler als direkt am Tisch erfolgen kann und von welchen Faktoren diese Ergebnisse beeinflusst werden können. McAdams und Brewster konnten in einer Studie zeigen, dass bei der Eingabe von Texten am Tabletop über eine virtuelle Tastatur signifikant weniger Fehler auftraten, als bei der Eingabe über eine herkömmliche Tastatur, über ein Mobiltelefon mit Touchscreen sowie über ein Mobiltelefon, das über Touchscreen und zusätzliches Feedback verfügt.²⁴³ Die Genauigkeit der Eingaben sollte jedoch nicht als alleiniges Merkmal für die Eignung einer Texteingabemethode herangezogen werden, wenn vor allem auch Geschwindigkeit und Benutzerzufriedenheit eine Rolle spielen. Letztlich muss sich zeigen, in welchen Szenarien sich Tabletops und externe Geräte wie Smartphones sinnvoll ergänzen können. Die im Rahmen dieser Arbeit konzipierte und prototypisch umgesetzte Anwendung kann an dieser Stelle als eine Ausgangsbasis für derartige Fragen verstanden werden. In den nachfolgenden Kapiteln soll nun schließlich auf die Konzeption und Implementierung des *Multi/Touch/Device MindMappers* eingegangen sowie erste Erfahrungen aus dem Testbetrieb diskutiert werden.

242 Vgl. Scott, Grant & Mandryk (2003, S. 6), vgl. Wallace et al. (2009, S. 576), vgl. Shoemaker & Inkpen (2001, S. 522), vgl. Hinrichs et al. (2007, S. 106).

243 Vgl. McAdam & Brewster (2009, S. 509 f.).

5 Anwendungskonzept

Die im Rahmen dieser Arbeit konzipierte und prototypisch umgesetzte Anwendung *Multi/Touch/Device MindMapper* soll als *proof-of-concept* für das Brainstorming und Mind-Mapping in einem Multi-Touch- und Multi-Device-Kontext dienen, wobei ein besonderes Augenmerk auf bestimmte Designziele, die im Kontext von Tabletop-Anwendungen und Multi-Device-Interaktion von Relevanz sind, gelegt werden soll. Dieses Kapitel beschreibt die Entwicklung des Anwendungskonzepts von der Aufgabenanalyse über die konkreten Anforderungen, die das System erfüllen soll, bis hin zu tatsächlichen Lösungsansätzen für bestimmte Designziele und Implementierungsdetails. Das Kapitel wird durch die Vorstellung eines ersten Prototyps in Form von papierbasierten Skizzen für einzelne Screens und Benutzerschnittstellenelemente abgeschlossen. Im nachfolgenden Kapitel 6 folgen schließlich Details zur Implementierung, wie die Architektur des Systems und die Beschreibung zentraler Programmklassen sowie die visuelle Umsetzung von Programmelementen. Die Strukturierung der folgenden Erläuterungen orientiert sich demnach grob an dem im Rahmen des Projekts eingehaltenen Software-Engineering-Prozess, welcher sich wiederum auf ein allgemeines Wasserfallmodell, bestehend aus den iterativen Phasen Anforderungserhebung, Analyse, Entwurf, Codieren, Testen und Betrieb abbilden lässt (Abbildung 27).

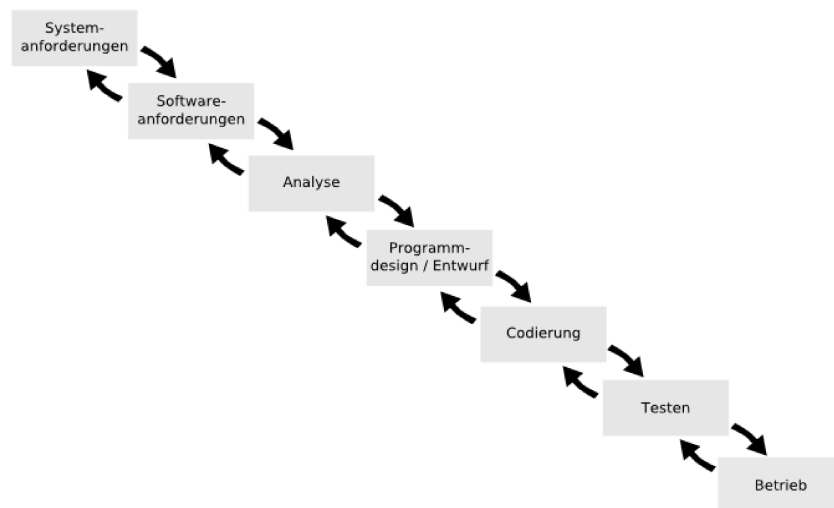


Abbildung 27: Iterative Projektschritte beim Wasserfallmodell (nach Royce 1987, S. 330).

5.1 Anforderungsbeschreibung

Ziel der Arbeit ist es, eine Anwendung zu konzipieren und umzusetzen, die das Brainstorming und Mind-Mapping in einem Multi-Device-Kontext an einem multitouchfähigen Tabletop in Kombination mit Smartphones ermöglicht. Zunächst stellt sich die Frage, welche Aufgaben eine derartige Anwendung unterstützen soll und welche Anforderungen dieses System in der Folge erfüllen können muss.

5.1.1 Aufgabenanalyse

Das primäre Einsatzszenario für den *Multi/Touch/Device MindMapper* ist in einem Meeting zur Ideenfindung oder Problemlösung zu sehen, in dessen Rahmen kollaborativ Ideen gesammelt und strukturiert werden müssen. Dies kann beispielsweise im Rahmen der Entwicklung eines neuen Produkts, der Planung eines Events oder auch bei der gemeinsamen Erstellung von Vorträgen der Fall sein. Während der Tabletop in diesem Kontext primär zur Unterstützung von Gruppenarbeitsprozessen, das heißt der Visualisierung, Koordination und Kommunikation von Ideen und deren Beziehungen zueinander dienen soll, schaffen zusätzliche Smartphones private Räume, in denen einzelne Gruppenmitglieder ungestört Informationen akquirieren sowie, darauf aufbauend, neue Ideen erstellen können, um diese schließlich auf die Tabletopoberfläche zu transferieren. Da jedoch nicht immer gewährleistet sein kann, dass jeder Benutzer ein Smartphone zur Verfügung hat, muss das Erstellen von digitalen Notizen am Tabletop ebenfalls möglich sein. Um Koordination und Kommunikation in der Gruppe zu unterstützen und das Potential der direkten Interaktion mit dem Tabletopsystem auszunutzen, werden Berührungsgesten zum Einsatz kommen und die einzelnen Elemente frei positionierbar und rotierbar sein. Da einzelne Gruppenteilnehmer eventuell bereits im Rahmen der Vorbereitung zum Meeting Ideen sammeln wollen, muss die Anwendung am Smartphone das Speichern von Ideen ermöglichen. Um jedoch den Fokus nicht zu sehr von der kollaborativen Aufgabe weg zu lenken und die *Awareness* innerhalb der Gruppe nicht zu sehr zu gefährden, soll die Funktionalität der Smartphone-Anwendung auf für die Individualarbeit zentrale Aspekte beschränkt bleiben. Die dem eigentlichen Brainstorming nachfolgende Selektion und Strukturierung von geeigneten Ideen ist idealerweise für alle Gruppenteilnehmer sichtbar und wird deshalb zunächst nur am Tabletop ermöglicht werden. Hierzu muss die Anwendung das Löschen und Editieren von Ideen ebenso unterstützen wie den Aufbau einer Mind-Map durch das Verbinden von ein-

zelen Ideen. Um die Ergebnisse des Meetings auch in nachfolgenden Sitzungen zur Verfügung zu haben, muss das Speichern und Laden des Arbeitsfortschritts am Tabletop ebenso möglich sein.

5.1.2 Funktionale Anforderungen

Aus dieser ersten, groben Aufgabenanalyse lassen sich sowohl für die Anwendung am Multi-Touch-Tisch als auch für die Smartphone-Applikation bereits grundlegende funktionale Anforderungen ableiten. Die Tabletop-Anwendung sollte demnach folgende grundlegenden Anforderungen erfüllen:

- Erstellen von Ideen
- Freies Verschieben und Rotieren der Ideen
- Bearbeiten von Ideen
- Erstellen von gerichteten Verbindungen zwischen Ideen zur Erstellung von Mind-Map-Strukturen
- Löschen von Ideen und Relationen
- Speichern der aktuellen Ideensammlung / Mind-Map
- Öffnen einer gespeicherten Ideensammlung / Mind-Map
- Minimieren und Schließen der Anwendung

Die Anwendung am Smartphone sollte folgende grundlegende Anforderungen erfüllen:

- Erstellen von Ideen
- Speichern von Ideen
- Löschen von Ideen
- Verbindung mit dem Tabletop
- Transfer einer Idee auf den Tabletop
- Unterstützung von Parallelaufgaben (Möglichkeit des parallelen WWW-Browsing etc.)

5.2 Implementierungskonzept

Über die von einer konkreten Umsetzung unabhängigen Funktionsanforderungen hinaus wird in der Folge die Wahl der jeweiligen Entwicklungsframeworks konkretisiert sowie die Kommunikation zwischen Tabletop und Smartphones näher erläutert.

Auswahl der Entwicklungsplattformen und -frameworks

Im Bereich der Smartphoneentwicklung dominieren derzeit neben Plattformen wie *Blackberry OS* oder *Windows Mobile* vor allem zwei Betriebssysteme den Markt: Googles *Android*-Plattform (53 Prozent Marktanteil) und Apples *iOS* (29 Prozent Marktanteil).²⁴⁴ Neben der Verbreitung der *Android*-Plattform und deren Unterstützung durch zahlreiche unterschiedliche Hersteller von Smartphones ist ein weiterer Vorteil der Entwicklung von Applikationen für dieses Betriebssystem die Tatsache, dass die Plattform unter *Open Source*-Lizenz (*Apache 2.0*, *GNU GPL 2*) steht und mit einem umfangreichen *Application Programming Interface* (API) sowie diversen Entwicklungstools eine flexible Entwicklungsumgebung zur Verfügung stellt. Die auf Java basierende Entwicklung von Anwendungen für *Android* Smartphones ermöglicht somit eine einfache und schnelle Umsetzung von (prototypischen) Anwendungen, weshalb im Rahmen dieser Arbeit die Smartphonekomponente des *Multi/Touch/Device MindMappers* als *Android*-Applikation umgesetzt werden soll.

Das optische Erkennungs- und Trackingsystem des im Rahmen der Arbeit eingesetzten *Evoluce ONE*-Tabletops unterstützt sowohl *Windows 7-Touch Events*²⁴⁵ als auch das *TUIO*-Protokoll.²⁴⁶ *TUIO* ist ein quelloffenes Framework, das ein Protokoll sowie eine API für interaktive Oberflächen, die *Tangibles* und Multi-Touch-Interaktion unterstützen, bereitstellt. Das *TUIO*-Protokoll ermöglicht die abstrakte Beschreibung von Touch-Events und Informationen über Objekte auf der Tischoberfläche und stellt somit eine Kommunikationsschnittstelle zwischen dem Trackingsystem (Kamerahardware und bildverarbeitende Software) sowie der jeweiligen *TUIO*-Client-Anwendung dar (Abbildung 28).²⁴⁷ Inzwischen sind bereits zahlreiche quelloffene Anwendungsframeworks für die Entwicklung von Multi-Touch-Anwendungen verfügbar, von denen die Mehrheit mit dem *TUIO*-Proto-

²⁴⁴ Vgl. Eddy (2011).

²⁴⁵ *Windows 7 Touch*: <http://msdn.microsoft.com/en-us/windows/ee633448.aspx>

²⁴⁶ *TUIO*: <http://www.tuio.org/>

²⁴⁷ Vgl. Tuio.org (2011).

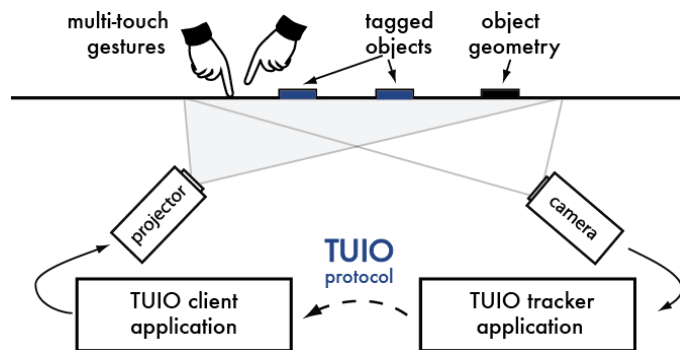


Abbildung 28: Schematische Darstellung des *TUIO*-Protokolls in einem Tabletop-System (Tuio.org 2011).

koll arbeitet. Eine nennenswerte Ausnahme stellt an dieser Stelle das *Microsoft Surface SDK*²⁴⁸ dar, welches lediglich *Windows 7*-Multi-Touch-Events unterstützt.²⁴⁹ Sowohl das *Surface SDK* (WPF/C#, proprietär) als auch auf dem *TUIO*-Protokoll aufbauende Frameworks wie *PyMT* (Python, GNU LGPL), *libTISCH* (C++, GNU LGPL), *MT4j* (Java, GNU GPL) oder *GestureWorks* (Flash, proprietär)²⁵⁰ stellen meist bereits grundlegende Gesten zur Manipulation von Objekten sowie einfache Benutzerschnittstellenelemente, wie virtuelle Tastaturen oder Buttons, zur Verfügung und erlauben somit eine effiziente Entwicklung von prototypischen Anwendungen für Tabletops.

Für die Entwicklung der im Rahmen dieser Arbeit umgesetzten Brainstorming und Mind-Mapping-Anwendung am Tabletop soll schließlich das *MT4j*-Framework zum Einsatz kommen. Dieses unterstützt bereits die Erkennung grundlegender Berührungaktionen wie *Single* und *Double Tap* sowie Manipulationsgesten wie *Drag and Drop* oder das Skalieren und Rotieren von Elementen. Darüber hinaus ermöglicht das Framework die Erkennung von symbolischen Unistroke-Gesten und stellt einige Multi-Touch-Widgets, wie eine virtuelle Bildschirmtastatur, scrollbare Listen oder Buttons, zur Verfügung. Eigene Widgets können unter Zuhilfenahme bestehender Multi-Touch-Primitive wie Rechtecke, Ellipsen und Linien ebenso wie eigens definierte Gesten entwickelt werden. Weitere Aspekte des Frameworks sollen schließlich in Kapitel 6 im Rahmen der Erläuterung der für die Anwendung relevanten technischen Grundlagen vorgestellt werden. Basierend auf Java ermöglicht es letztlich auch eine prinzipiell plattformunabhängige

²⁴⁸ *Microsoft Surface SDK*: <http://www.microsoft.com/download/en/details.aspx?id=26716>

²⁴⁹ Vgl. Kammer et al. (2010).

²⁵⁰ *PyMT*: <http://pymt.eu/>, *LibTISCH*: <http://tisch.sourceforge.net/>, *MT4j*: <http://www.mt4j.org/>, *GestureWorks*: <http://gestureworks.com/>

Implementierung der Tabletop-Anwendung sowie die komfortable Umsetzung einer Kommunikationsschnittstelle zur ebenfalls in Java implementierten *Android*-Applikation.

Kommunikation zwischen den Geräten

Für die drahtlose Kommunikation zwischen den Smartphones der Benutzer und dem Tabletop bieten sich vor allem zwei Möglichkeiten an: die Verbindung der Geräte und der Transfer von Daten über das Netzwerk oder über Bluetooth. Moderne Smartphones unterstützen in der Regel sowohl den drahtlosen Zugang zum Internet als auch die Kommunikation mit anderen Geräten über das Bluetoothprotokoll. Da die Umsetzung einer einfachen Client-Server Architektur zur Kommunikation über Bluetooth sowohl für *Android* als auch am Tabletop über Javabibliotheken sehr effizient möglich ist und das Systemdesign weder eine hohe Reichweite der Verbindung noch den Transfer von umfangreichen Daten erfordert, soll die Kommunikation zwischen Smartphones und Tabletop in der prototypischen Anwendung zunächst lediglich über Bluetooth umgesetzt werden.

5.3 Designziele und Lösungsansätze

Neben den grundlegenden Funktionen, welche die Anwendung auf dem Tabletop und auf dem Smartphone bereitstellen muss, lassen sich für die Gestaltung dieser Benutzerschnittstellen zusätzlich einige Designziele sowie diesbezüglich konkrete Lösungsansätze formulieren. Für den am Tabletop ausgeführten Teil des *Multi/Touch/Device MindMappers* wurden diese vor allem aus den zuvor identifizierten Forschungs- und Problemfeldern und diversen, dabei referenzierten Veröffentlichungen zur Entwicklung von kollaborativen Benutzerschnittstellen mit Guidelinecharakter abgeleitet.²⁵¹ In der Folge wird auf das allgemeine Screenkonzept sowie auf konkrete Aspekte der Benutzerschnittstellengestaltung eingegangen und ein Gestenalphabet für die Interaktion an Tabletop und Smartphone vorgeschlagen werden.

5.3.1 Screenkonzept

Screenkonzept Tabletop

Das für den Tabletop umgesetzte Screenkonzept folgt dem Designziel einer einfachen und intuitiv bedienbaren Benutzeroberfläche, welche die Interaktion mit Berührungsgesten erlaubt. Die Arbeitsoberfläche soll dabei von einer einzigen Ansicht ohne weitere Fenster gebildet werden, auf der alle Elemente mit Multi-Touch-Gesten frei positioniert und rotiert werden können. Weitere, auf einzelne Inhalte bezogene Funktionen, wie das Erstellen und Löschen von Ideen (die durch Notizzettel repräsentiert werden) und Relationen, sollen mit Hilfe von symbolischen Gesten ausgeführt werden, um die direkte Interaktion mit dem System zu fördern. Funktionen, die sich auf die gesamte Anwendung beziehen (Hilfe, Speichern, Laden, Schließen etc.), sollen in kreisförmigen Menüs repräsentiert und selektiert werden. Die Darstellung der einzelnen Menüelemente erfolgt dabei nicht über Text, sondern über grafische Icons. Die radiale Ausrichtung und Darstellung mittels Icons unterstützt an dieser Stelle das Designziel, Bildelemente für die flexible Interaktion aus allen Perspektiven zu optimieren. Zusätzliche Informationen, wie Erläuterungen zu den möglichen Interaktionsgesten, werden in eigenen Overlays dargestellt, die ebenfalls frei positioniert und rotiert werden können. Für die Erläuterungen zu den einzelnen Gesten soll es die Mög-

²⁵¹ Vgl. Scott, Grant & Mandryk (2003), vgl. Wallace & Scott (2008), vgl. Apted, Collins & Kay (2009), vgl. Microsoft (2009), vgl. Remy, Weiss & Borchers (2010).

lichkeit der mehrfachen Anzeige geben, damit sie auch von mehreren Nutzern gleichzeitig rezipiert werden können. Das Design der einzelnen Bildelemente sollte zudem auf unterschiedliche Bildschirmauflösungen anpassbar sein.

Screenkonzept Smartphone

Die Bildschirmoberfläche an Smartphones ist, im Gegensatz zur Oberfläche an Tabletops, durch die variable Hardware und der generell geringeren Dimensionen mobiler Geräte stark eingeschränkt, weshalb die einzelnen Funktionen auf unterschiedliche Screens aufgeteilt werden müssen. Da sich die Funktionen am Smartphone auf das Erstellen und Verwalten von Ideen, deren Transfer und das Herstellen einer Verbindung zum Tabletop beschränken, bietet sich hier die Aufteilung in a) Ideentransfer, b) Ideenverwaltung und c) Funktionen zur Verbindung mit dem Tabletop an. Für einzelne Benutzerschnittstellenelemente kann und wird an dieser Stelle auf standardisierte Widgets der gewählten Zielplattform *Android* zurückgegriffen werden.

5.3.2 Gestenalphabet

Sowohl für die Interaktion am Tabletop als auch am Smartphone sollen Single- und Multi-Touch-Gesten angewandt werden, um das Designziel der direkten Interaktion mit der Benutzeroberfläche zu ermöglichen. In den Entwurf eines geeigneten Gestenalphabets sind unterschiedliche Aspekte einzubeziehen. So sollte zum einen gewährleistet sein, dass Aktionen, die sowohl am Tabletop als auch am Smartphone ausgeführt werden können, an beiden Systemen durch die gleichen Gesten repräsentiert werden. Für die Interaktion mit Smartphones haben sich bereits bestimmte Gesten durchgesetzt, wie der *Single Tap* zur Auswahl von Elementen oder eine *Flick*-Geste zum Scrollen durch Listen. Häufig angewandt werden vor allem Gesten, welche bisherige Mausaktionen abbilden, wie *Single Tap*, *Double Tap* oder *Drag and Drop*. Neben dem Blick auf jene Gesten, die bis zu einem gewissen Grad als bereits etabliert betrachtet werden können, lohnt sich, in Ermangelung eigener Feldstudien, ein Blick auf bisherige Tabletopsysteme im Kontext von Brainstorming und Mind-Mapping und die dabei umgesetzten Berührungsgesten sowie auf gegebenenfalls bereits existierende Studien zu geeigneten Gesten, speziell bei Erstellung und Manipulation von baumartigen Strukturen. Letztlich hängt die konkrete Umsetzbarkeit auch vom gewählten Anwendungsframework ab.

Gestenalphabet Tabletop

Während bei zahlreichen der bereits vorgestellten Tabletop-Anwendungen zum Brainstorming und Mind-Mapping lediglich Mausaktionen auf Berührungsgesten abgebildet wurden, kommt bei Kims *Brainstorm*-System auch eine Vielzahl weiterer Gesten zum Einsatz:²⁵² Die Erstellung einer neuen Idee in Form eines Notizzettels erfolgt über das Zeichnen eines Rechtecks bzw. eines Kreises auf der Tabletopoberfläche. Das Beschreiben eines neuen Notizzettels kann nach Tippen in dessen Mitte erfolgen. Neben dem üblichen Bewegen von Notizzetteln über *Drag and Drop* können Ideen auch mit Hilfe einer *Flick*-Geste über den Tisch „geworfen“ werden. Mehrere Ideen lassen sich zu Ideenclustern zusammenfassen, indem sie mit einer Lasso-Geste (Zeichnen eines kreisförmigen Bereichs um mehrere Elemente) selektiert werden. Diese Ideencluster können wiederum miteinander verbunden werden, wenn eine Linie zwischen zwei Clustern gezeichnet wird. Das Löschen dieser Verbindung geschieht durch eine Zick-Zack-Geste über der Verbindungslinie.

Einige dieser Gesten finden sich auch in einer 2009 von Frisch, Heydekorn und Dachselt veröffentlichten Studie zu Finger- und Stiftgesten zum Editieren von Diagrammen an interaktiven Oberflächen.²⁵³ Ziel der Untersuchung war es, ein Gestenset für das Editieren von Knoten-Kanten Diagrammen auf der Basis von Benutzervorschlägen zu erstellen. Für diverse Aufgaben, wie das Erstellen, Bewegen, Selektieren und Löschen von Knoten sowie die Verbindung einzelner Knoten mit gerichteten und ungerichteten Kanten sollten die Testpersonen geeignete Gesten vorschlagen. Diese durften mit einer oder zwei Händen sowie mit einem Stift ausgeführt werden. Bei der Mehrheit der dabei erfassten Gesten handelte es sich schließlich um mit einer Hand ausgeführte Gesten, wobei auffiel, dass die Benutzer kaum zwischen der Interaktion mit dem Finger oder mit dem Stift unterschieden und überwiegend gleiche Gesten in beiden Konditionen ausführten. Das daraus abgeleitete Gestenset umfasst für die einzelnen Aufgaben unter anderem folgende Gesten:²⁵⁴

- **Knoten erstellen:** Wiederholtes Tapping auf der Oberfläche (*Double Tap*), Zeichnen des Umrisses des Knotens auf der Oberfläche (Rechteck, Kreis) oder Kopieren eines bestehenden Knotens durch Selektion und Ziehen

²⁵² Vgl. Kim (2006, S. 13 ff.).

²⁵³ Vgl. Frisch, Heydekorn & Dachselt (2009).

²⁵⁴ Vgl. Frisch, Heydekorn & Dachselt (2009, S. 154).

- **Ungerichtete Kante erstellen:** Zeichnen einer Linie zwischen zwei Knoten mit einem oder zwei Fingern, sequentielles Tapping zweier Knoten
- **Gerichtete Kante erstellen:** Zeichnen eines Pfeils von einem Knoten zu einem anderen Knoten oder Zeichnen einer Linie zwischen zwei Knoten mit gleichzeitiger Berührung des Elternknotens
- **Auswahl eines einzelnen Knotens:** Wiederholtes Tapping auf einen Knoten (*Double Tap*) oder Zeichnen eines Rahmens um einen Knoten (Lasso-Geste)
- **Auswahl mehrerer Knoten:** Sequentielles Tapping mehrerer Knoten oder Zeichnen eines kreisförmigen Rahmens um mehrere Knoten (Lasso-Geste)
- **Bewegen eines einzelnen Knotens:** Berührung des Knotens und Ziehen mit Finger oder Stift (*Drag and Drop*-Geste)
- **Bewegen einer Gruppe (verbundener) Knoten:** Berührung eines Knotens der Gruppe und Ziehen mit Finger und Stift (*Drag and Drop* Geste)
- **Knoten löschen:** Ausführen einer Wisch-Geste über dem Knoten oder Herausziehen des Knotens aus dem Bildschirmbereich
- **Kanten löschen:** Ausführen einer Wisch-Geste über der Kante oder Herausziehen der Kante aus dem Bildschirmbereich

Aufbauend auf diesem speziell auf die Bearbeitung von Kanten-Knoten-Diagrammen ausgerichteten Gestenset, der im Rahmen weiterer Projekte umgesetzten Gesten, allgemeinen Empfehlungen zum Einsatz von Gesten an Smartphones und interaktiven Oberflächen sowie in Orientierung an den im Rahmen des *MT4j*-Framework bereits zur Verfügung gestellten Gesten wurde schließlich ein erstes Gestenalphabet für zentrale Aktionen und Funktionen innerhalb der Tabletopkomponente der *Multi/Touch/Device MindMapper*-Anwendung erstellt:²⁵⁵

- **Single Tap:** Selektion von Objekten, Funktionsauslösung, Texteingabe
- **Double Tap auf einer Idee:** Editieren des Ideentextes

²⁵⁵ Vgl. Wobbrock, Morris & Wilson (2009, S. 1089), vgl. Le Hong & Biesterfeld (2010, S. 40 ff.), vgl. Saffer (2009, S. 45 ff.).

- **Zeichnen eines Rechtecks/Kreises auf der Oberfläche:** Erstellen einer neuen Idee
- **Lasso-Geste:** Selektion von mehreren Ideen
- **Drag and Drop-Geste:** Bewegen eines Objektes
- **Flick-Geste:** schnelles Bewegen eines Objekts in eine Richtung
- **Rotations-Geste (Zwei Finger):** Rotieren eines Objekts
- **Zeichnen eines Pfeiles zwischen zwei Ideen:** Erstellen einer Relation zwischen zwei Ideen (gerichtete Verbindung von Eltern- zu Kindelement)
- **Zeichnen eines X auf einer Idee/Relation:** Löschen der Idee/Relation

Gestenalphabet Smartphone

Für das Gestenalphabet am Smartphone wird in erster Linie auf die bereits in der Plattform integrierte Gesteninteraktion zurückgegriffen. Diese umfasst unter anderem den **Single Tap**, der ebenso wie am Tabletop zur Auswahl von Objekten und zur Funktionsauslösung und Texteingabe dient, sowie die **Wisch-** bzw. **Flick-Geste** zum Scrollen durch Listeneinträge. Für den Transfer einer Idee auf den Tabletop bietet sich an dieser Stelle in Anlehnung an die am Tisch umgesetzte Notizzettelmetapher ebenfalls der Einsatz direkter Manipulation in Form einer Geste an, weshalb der Transfer einer Idee vom Smartphone auf den Tabletop schließlich über eine **Dragging-Geste** auf einen designierten Bereich der Benutzeroberfläche am Smartphone realisiert werden soll.²⁵⁶

5.3.3 Benutzerschnittstellengestaltung

Ein ausgereiftes visuelles Konzept wird aufgrund des lediglich prototypischen Charakters der umgesetzten Anwendung weniger im Fokus der Arbeit stehen. Dennoch soll versucht werden, beispielsweise Farbschemata am Tabletop ähnlich derer am Smartphone zu gestalten, um einen flüssigen Übergang vom Arbeiten am Smartphone zur Interaktion am Tabletop zu unterstützen. Im eigentlichen Fokus stehen an dieser Stelle vielmehr bestimmte Interaktionslösungen und die konkrete Umsetzung von Anforderungen für die jeweilige Hardware.

²⁵⁶ Vgl. Prante, Streitz & Tandler (2004, S. 50 f.), vgl. Streng et al. (2010, S. 790).

Während sich die Benutzerschnittstellengestaltung am Smartphone maßgeblich an bestehenden Guidelines zur Erstellung von Applikationen für die *Android* Plattform orientiert, zeigen die bereits in Kapitel 3.2 vorgestellten Problem- und Forschungsfelder, dass bei der Konzeption der Anwendung am Tabletop einige besondere Aspekte beachtet werden müssen. Die *Okklusion* von Bildelementen bzw. das *Fat-Finger*-Problem kann beispielsweise durch die Einhaltung von Mindestgrößen von Touchtargets umgangen werden. Alle Elemente, die auf Berührung reagieren sollen (bzw. deren für Berührung empfindliche Bereiche), darunter einzelne Ideen, Relationen zwischen Ideen, die Menüs und die darin auswählbaren Menüpunkte sowie Overlays, Dialoge und Buttons, müssen groß genug sein, um mit einem Finger selektiert werden zu können, jedoch nicht so groß, dass die Gefahr der Überfrachtung der Arbeitsfläche besteht (siehe Kapitel 3.2.2). Visuelles Feedback wird durch das *MT4j*-Framework durch die Möglichkeit der Visualisierung der Berührungspunkte und der Veränderung von Buttons und Listenelementen bei Selektion bereitgestellt. Um den Benutzern beim Einsatz von Unistroke-Gesten zusätzliches Feedback zu geben, soll die gezeichnete Form entsprechend einer erfolgreichen Erkennung farblich dargestellt werden. Wird eine Geste erkannt und kann die damit verknüpfte Aktion ausgeführt werden, erfolgt eine kurze Darstellung der Geste in grün. Bei erfolgreicher Erkennung einer Geste ohne die Möglichkeit einer Aktion (aufgrund von Constraints oder der falschen Platzierung der Geste) wird sie in gelb dargestellt. Wird die Geste nicht erkannt, ist sie kurz in rot zu sehen.²⁵⁷

Die Problematik eines fehlenden *Hover*-Zustandes wird dahingehend abgemildert, dass die Selektion von Elementen wie Buttons oder Listenelementen nicht bereits beim Registrieren einer Berührung, sondern erst beim Wiederwegnehmen des Fingers erfolgt, wodurch der Benutzer eine fälschliche Auswahl noch korrigieren kann. Ein zentrales Designziel ist schließlich auch die Gestaltung der Benutzerschnittstelle für eine flexible Interaktion von allen Seiten des Tabletops, weshalb alle Bildelemente frei positionierbar und rotierbar sein sollen. Einzelne Ideen, die in Form von Notizzetteln dargestellt werden, beinhalten den jeweiligen Ideentext darüber hinaus nicht nur in einer Ausrichtung, sondern zusätzlich gespiegelt an der Achse, um die Lesbarkeit aus verschiedenen Perspektiven zu gewährleisten, wobei einzelne Ideen selbstverständlich auch immer ro-

²⁵⁷ Der Einsatz von Farben zur Unterscheidung von Zuständen ist selbstverständlich aufgrund von kulturellen Konventionen sowie der Möglichkeit der Farbenblindheit von Benutzern nicht ganz unproblematisch, jedoch konnte im Rahmen der Arbeit keine umfangreichere Feedbackfunktionalität umgesetzt werden. Zu möglichen Erweiterungen sei an dieser Stelle auf die Ausführungen in Kapitel 5.4 verwiesen.

tiert werden können.²⁵⁸ Schließlich soll die Anwendung dem Benutzer diesbezüglich insofern entgegenkommen, als bei der Erstellung einer neuen Idee am Tabletop diese je nach ihrer Position auf der Oberfläche automatisch ausgerichtet wird (siehe Kapitel 3.2.3). Genauso sollen Overlays bei deren Aufschalten nach der Position des aufrufenden Menüs auf der Arbeitsfläche ausgerichtet werden. Neue Ideen werden nach dem Transfer von einem Smartphone auf den Tabletop in der Mitte der Arbeitsfläche positioniert, da hier keine Annahmen über die Position des Benutzers am Tisch getroffen werden können.

Funktionen, die sich auf die gesamte Anwendung beziehen und somit nicht von dem vorgestellten Gestenalphabet abgedeckt werden, können in kreisförmigen, frei positionier- und rotierbaren Menüs selektiert werden. Neben der in den Anforderungen definierten Funktionalität zum Laden und Speichern von Ideensammlungen / Mind-Maps sowie zum Schließen und Minimieren der Anwendung soll auch eine rudimentäre Hilfe zu den verwendbaren Gesten zur Verfügung gestellt werden und ein Informationsoverlay zu aktuellen Bluetoothverbindungen eingeblendet werden können. Die Darstellung dieser einzelnen Menüpunkte erfolgt über Icons, um auch hier die Bedienung aus allen Perspektiven zu erleichtern. Die Eingabe von Texten wird schließlich über in *MT4j* bereits vordefinierte virtuelle Tastaturen ermöglicht, die jeweils an das zu bearbeitende Textfeld gekoppelt sind.

Neben diesen Aspekten zur generellen Gestaltung der Benutzerschnittstelle und einzelnen Elementen sollen auch Designziele realisiert werden, die Einfluss auf die kollaborative Arbeit nehmen. So scheint es sinnvoll, zur Förderung von *Awareness* innerhalb der Gruppe (siehe Kapitel 2.4) zusätzliches visuelles Feedback bei der Erstellung einzelner Notizzettel sowie insbesondere beim Transfer von Ideen von externen Geräten zu realisieren, weshalb die neuen Ideen mit einer Animation auf dem Bildschirm dargestellt werden sollen, wobei der Transfer einer Idee von einem Smartphone auf den Tabletop stärker hervorgehoben wird als die Erstellung eines neuen Notizzettels direkt am Tisch. Die freie Positionierung und Ausrichtung von Elementen unterstützt darüber hinaus die individuelle Segmentierung der Arbeitsoberfläche durch die Benutzer im Sinne der von Scott, Carpendale und Inkpen vorgestellten Theorie zur Territorialität an Tabletop-schnittstellen, wobei die Smartphones diesbezüglich zusätzliche persönliche Bereiche zur Verfügung stellen.²⁵⁹ Letztlich wird auf globale Aktionen, die den Status der gesamten Anwendung verändern, so weit wie möglich verzichtet und vor

²⁵⁸ Vgl. Buisine et al. (2007, S. 25).

²⁵⁹ Vgl. Scott, Carpendale & Inkpen (2004).

dem Ausführen von Aktionen, die – wie das Schließen der Anwendung – Einfluss auf alle Benutzer haben, eine erneute Bestätigung in einem Dialog-Overlay verlangt. Dialoge bilden die einzige Ausnahme bezüglich der freien Positionierung und Ausrichtung auf der Benutzeroberfläche – da sie die Aufmerksamkeit aller Benutzer erfordern, sollen sie zentral und mit fester Ausrichtung in der Mitte der Anwendung eingeblendet werden und dabei einen Großteil der Bildschirmoberfläche einnehmen. Um dennoch Lesbarkeit aus mehreren Perspektiven zu gewährleisten, sollen der Informationstext und etwaige Buttons hier ebenfalls an der Achse gespiegelt werden. Derartige Dialoge können zusätzlich zur Anzeige von die gesamte Anwendung betreffenden Informationen, wie dem Bluetooth-Status oder dem Erfolg beim Laden oder Speichern einer Ideensammlung, verwendet werden.

Diese Designziele, welche die grundlegenden funktionalen Anforderungen unterstützen, können im Rahmen der Arbeit umgesetzt werden – selbstverständlich wären an dieser Stelle zahlreiche zusätzliche Erweiterungen denkbar. Einige dieser Möglichkeiten sollen schließlich im folgenden Kapitel 5.4 kurz skizziert werden.

5.4 Erweiterungsmöglichkeiten

Neben den zuvor angeführten funktionalen Anforderungen sowie den darüber hinaus gehenden Designzielen sind im Rahmen einer Anwendung zum Brainstorming und Mind-Mapping am Tabletop mit Smartphones weitere Erweiterungsmöglichkeiten denkbar, die jedoch im Rahmen dieser Arbeit, nicht zuletzt aus zeitlichen Gründen, nicht umgesetzt werden können.

Zunächst wäre neben der Erstellung und Verwaltung von Ideen sowie deren Transfer über Bluetooth weitere Funktionalität am Smartphone denkbar. An dieser Stelle müsste jedoch darauf geachtet werden, dass die Möglichkeiten der Interaktion mit dem externen Gerät das Arbeitsszenario nicht derart dominieren, dass die eigentliche Kollaboration und Kommunikation um den Tisch herum darunter leiden. Als minimale Ergänzung wäre hier eine Anzeige der bisher transferierten Ideen in Relation zu allen Ideen auf dem Tabletop denkbar, um dem Benutzer Rückmeldung zu seiner Beteiligung an der Sitzung zu geben und ihn gegebenenfalls zu einer aktiveren Teilnahme anzuregen. Neben der grundlegenden Anforderung, einzelne Ideen auf den Tabletop übertragen zu können, stellt sich die Frage, ob nicht auch der Transfer mehrerer (selektierter) Ideen auf einmal sinnvoll wäre. Des Weiteren wäre vorstellbar, dass jeder Benutzer auch eine Kopie der gesamten Ideensammlung / Mind-Map auf seinem Smartphone speichern kann, um diese in einem nachfolgenden Meeting auf den Tabletop transferieren zu können. Weitere Editiermöglichkeiten der gesamten Mind-Map am Smartphone wären an dieser Stelle vermutlich bereits der Kollaboration am Tisch abträglich, was jedoch auch im Rahmen einer evaluativen Studie von Interesse sein könnte.

Sowohl am Smartphone als auch am Tabletop wären neben der Erstellung von Ideen in Form von Text auch das Anfertigen von Skizzen denkbar. Über die Texteingabe an der virtuellen Tastatur hinaus könnte auch die handschriftliche Eingabe von Text einen Mehrwert bieten. Schließlich sind bezüglich der allgemeinen Kommunikation zwischen Smartphone und Tabletop einige weitere Szenarien denkbar – so ist in dem im Rahmen dieser Arbeit vorgestellten Anwendungskonzept zunächst nur eine unidirektionale Kommunikation von Smartphone zum Tabletop in Form des Transfers von kurzem Text vorgesehen. Um das Potential der Multi-Device-Interaktion voll auszuschöpfen, wäre vorstellbar, dass Ideen auch wieder zurück auf ein Smartphone transferiert werden können. Hierfür müssten jedoch einzelne externe Geräte am Tabletop identifizierbar sein, was beispielsweise durch einen personalisierten Bereich, das heißt einen mit dem ex-

ternen Gerät assoziierten Bereich, in den Objekte zum Transfer gezogen werden können, erreicht werden könnte.²⁶⁰ Wenn es zudem möglich wäre, einzelne Notizzettel dem jeweiligen Smartphone, von dem diese Ideen transferiert wurden, zuzuordnen, könnten bei weiterer Manipulation eines Zettels (Löschen, Editieren, Verbinden mit einem weiteren Notizzettel) Rückmeldungen an den jeweiligen Benutzer über sein Smartphone gegeben werden.²⁶¹ Eine derartige Zuordnung würde schließlich auch erlauben, einzelne Zettel beispielsweise farblich zu markieren um die Kontribution einzelner Benutzer hervorzuheben und so gegebenenfalls den bei Gruppenarbeit auftretenden Trittbrettfahreneffekt (siehe Kapitel 2.2) zu verhindern.

Eine weitere unmittelbar mögliche Erweiterung wäre zudem die Möglichkeit der Skalierung von Bildelementen durch eine entsprechende Geste. Diese Funktionalität wurde im Rahmen des vorgestellten Gestenalphabets zunächst nicht definiert, da dies für die grundlegenden Anforderungen an die Anwendung als zunächst nicht relevant und womöglich auch als störend in dem speziellen Aufgabenkontext „Ideengenerierung und -strukturierung“ erachtet wird. Die Erweiterung der Anwendung um die Möglichkeit der Skalierung von Ideen, Menüs oder Overlays ist jedoch jederzeit möglich.

Ebenfalls relativ einfach umsetzbar wäre erweitertes Feedback bei der Erkennung von Unistroke-Gesten. Im Rahmen der Arbeit soll durch Farbfeedback zumindest eine Art von Rückmeldung bezüglich des Erfolgs der Gestenausführung umgesetzt werden, an dieser Stelle wären jedoch gegebenenfalls zusätzliche Informationen sinnvoll, um den Benutzer darüber in Kenntnis zu setzen, warum eine Funktion, wie zum Beispiel das Erstellen einer Relation, nicht ausgeführt wird. Sei es aufgrund der Tatsache, dass die ausgeführte Geste vom System nicht erkannt wurde oder weil bestimmte Vorbedingungen zur Ausführung der Aktion nicht erfüllt sind, zum Beispiel, weil eine Idee innerhalb einer Mind-Map nur mit einer weiteren übergeordneten Idee verbunden werden darf. Dies könnte durch zusätzliche Overlays mit textueller oder ikonischer Information an der Position, an der der Benutzer die Geste ausgeführt hat, realisiert werden.

Obwohl die Ausführung von globalen Systemaktionen in der Regel in einem Dialog bestätigt werden muss, könnte hier der Aspekt der Kollaboration weiter fokussiert werden, indem eine globale Aktion nur dann ausgeführt werden kann, wenn alle Benutzer am Tisch dies bestätigen. Die Umsetzung einer derartigen

²⁶⁰ Vgl. Chehimi & Rukzio (2010, S. 443 f.).

²⁶¹ Anregung aus dem Auditorium im Rahmen der Vorstellung der Arbeit im Oberseminar des Faches Informationswissenschaft im Sommersemester 2011.

Mechanik könnte jedoch problematisch sein, da ohne die Identifizierung von Benutzern oder das Wissen um die genaue Anzahl der Teilnehmer die Bedingung zur letztendlichen Ausführung einer Aktion schwer zu definieren sein dürfte.

Ein weiterer Aspekt ist das Löschen von Ideen. Die im Rahmen dieser Arbeit definierte Anforderung sieht letztlich das endgültige Löschen von Ideen mit nur einer Geste vor, ein Rückgängigmachen einer Löschaktion ist nicht möglich. Die Ergänzung der Anwendung um eine Art „Papierkorb“, in dem gelöschte Ideen gesammelt werden und bei Bedarf wieder auf die Arbeitsfläche transferiert werden können, könnte hier zur allgemeinen Benutzbarkeit des Systems beitragen. Alternativ wäre auch ein Warndialog vor dem endgültigen Löschen einer Idee denkbar. Während der *Multi/Touch/Device MindMapper* das Speichern einer Ideensammlung / Mind-Map am Tabletop ermöglichen soll, wäre ebenso eine Exportfunktion in Form einer Grafikdatei oder in Form von strukturierten Daten (XML) denkbar – letzteres könnte auch den Import von bereits strukturierten Daten ermöglichen.

Die prototypische Umsetzung der im Rahmen der Arbeit konzipierten Anwendung wird sich jedoch auf die in Kapitel 5 beschriebenen Anforderungen und Designziele beschränken. Diese erste konzeptuelle Beschreibung des Systems wurde in Form eines ersten Prototyps in Form von papierbasierten Skizzen umgesetzt, um erste Designlösungen für die Benutzerschnittstellen vorzuschlagen und im Kontext der durch die eingesetzten Frameworks und Plattformen umsetzbaren Möglichkeiten sowie im Rahmen des durch die jeweilige Hardware gegebenen Kontextes darzustellen. Das folgende Kapitel soll kurz auf die Umsetzung dieser Skizzen eingehen.

5.5 Anwendungsskizzen und Papierprototyp

Im Zuge der ersten Definition von Designzielen und Anforderungen an die Anwendung wurden zur Visualisierung bisheriger Ideen erste Skizzen für die Anwendung am Smartphone (unter Berücksichtigung der Guidelines zur Erstellung von *Android*-Applikationen) und eine rudimentäre Form eines Papierprototypen für die Anwendung am Tabletop (unter Berücksichtigung der Dimensionen des Tisches²⁶² sowie der bisherigen Überlegungen zu Designzielen) angefertigt. In der Folge soll die Umsetzung und Entwicklung dieser Skizzen dokumentiert werden.

5.5.1 Papierprototyp Tabletop

Um die für die einzelnen Bildelemente nötigen Dimensionen abschätzen zu können, wurden iterativ papierbasierte Skizzen erstellt, die eine erste Annäherung an die Benutzerschnittstelle am Tabletop visualisieren sollen. Zentrale Aspekte waren dabei die Abwägung zwischen der Mindestgröße von Elementen zur komfortablen Auswahl mit dem Finger sowie zur Gewährleistung der Lesbarkeit am Tisch und der Beschränkung der Größe von Elementen zur Verhinderung der Überfrachtung der Benutzerschnittstelle. Abbildung 29 zeigt die Elemente des Papierprototyps auf der Oberfläche des *Evoluce ONE*-Tabletopsystems.



Abbildung 29: Visualisierung von Designideen für die Tabletop-Anwendung in Form eines maßstabsgetreuen Papierprototyps.

262 Abmessungen des Bildschirms des *Evoluce ONE*: 1034 mm x 585 mm.

Ideen und Relationen

Für die Visualisierung von Ideen wurde in Anlehnung an die Notizzettelmetapher schließlich eine rechteckige Darstellung gewählt, in welcher der Text an der mittleren Achse gespiegelt wird, um die Lesbarkeit aus mehreren Perspektiven zu ermöglichen. Die Eingabe von Text soll dabei auch zweizeilig erfolgen können, wobei eine größere Darstellung von kurzen Texten zur besseren Lesbarkeit sinnvoll scheint. Die Abmessungen eines digitalen Notizzettels sind an dieser Stelle mit 70 mal 50 mm der Versuch eines Kompromisses zwischen angemessener Textlänge, Lesbarkeit und Selektierbarkeit. Abbildung 30 zeigt, neben der finalen Darstellung, auch einige der ersten Ideen bezüglich der Größe von Ideenknoten und der Ausrichtung der Texte. Die Verbindung zwischen einzelnen Ideen wird schließlich durch einen Pfeil vom Eltern- zum Kindknoten visualisiert.

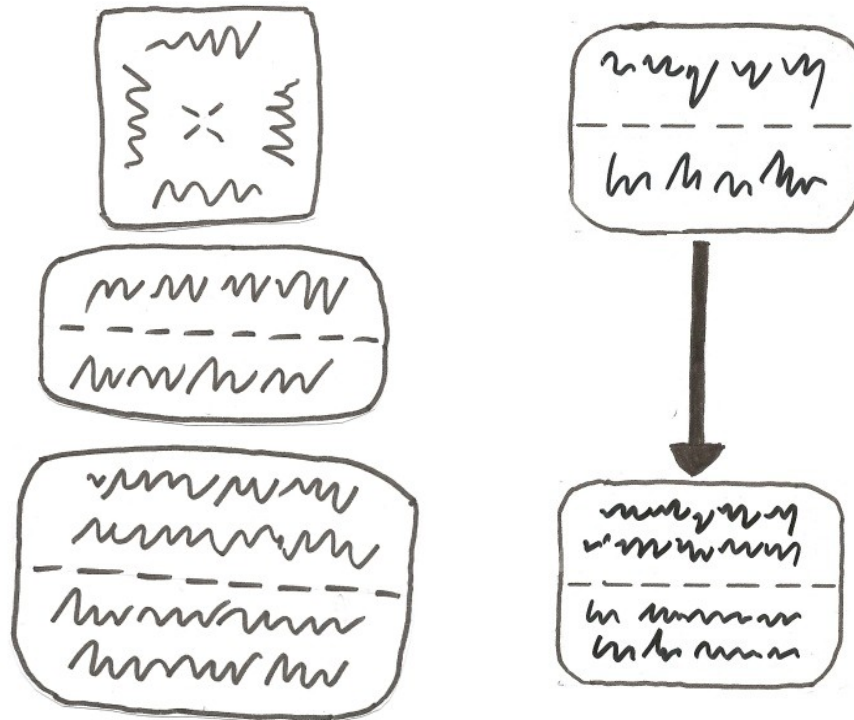


Abbildung 30: Skizzen zur Visualisierung von Ideen und Relationen auf dem Tabletop: erste Ideen, 55 x 55 mm, 85 x 40 mm und 90 x 60 mm (links, v.o.n.u.) und finale Skizze 70 x 50 mm (rechts).

Menüs

Für die Menüs wurde eine kreisförmige Darstellung gewählt, in denen die einzelnen Menüpunkte in Form von mit Icons versehenen Buttons radial um den Mittelpunkt des Menüs angeordnet werden. Da sich die Menüaktionen grob in allgemeine Anwendungsfunktionen (Laden, Speichern, Hilfe, Bluetooth) und Aktionen, die zur Manipulation des Anwendungsfensters dienen (Schließen, Minimieren und Verkleinern/Maximieren), einteilen lassen, scheint eine Separation in zwei Menüs sinnvoll, um einerseits eine gewisse Mindestgröße für die einzelnen Buttons in den Menüs zu ermöglichen und andererseits zu gewährleisten, dass die Menüs nicht so groß werden, dass die eigentlichen Inhalte der Anwendung optisch in den Hintergrund treten. Das Hauptmenü wird dabei größer (Durchmesser 80 mm) als das weniger häufig benutzte Fenstermenü dargestellt (Durchmesser 60 mm). Da die Buttons bereits den Großteil der Fläche der Menüs einnehmen, ist es nötig, zusätzlich Bereiche zu definieren, die zur Bewegung und zum Rotieren des Menüs verwendet werden können. Hierzu dienen schließlich der mittlere Bereich sowie an den Rändern befindliche ‚Griffe‘, welche durch die explizite Visualisierung der Möglichkeit des freien Platzierens und Rotierens des Menüs zusätzlich die *Affordanz* dieses Benutzerschnittstellenelements erhöhen sollen (Abbildung 31). Bei den Icons wurde versucht, allgemein verständliche bzw. etablierte Darstellungen für die einzelnen Aktionen zu wählen.

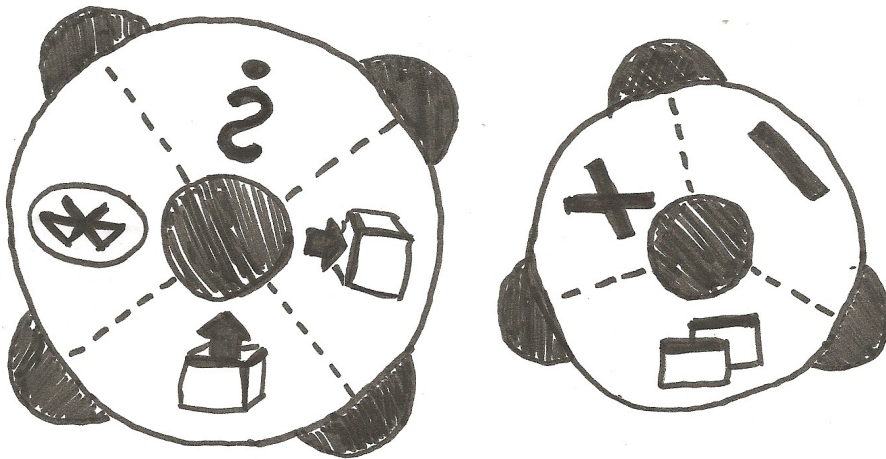


Abbildung 31: Skizzen zur Visualisierung der Menüs am Tabletop: Allgemeines Funktionsmenü, \varnothing 80mm (links) und Fenstermenü, \varnothing 60mm (rechts).

Overlays

Neben den eigentlichen Ideen und deren Verbindungen sowie den Menüs werden auch diverse Overlays Teil der Benutzerschnittstelle am Tabletop sein. Jede der vier Menüoptionen im Hauptmenü öffnet ein entsprechendes Overlay – beim Speichern einer Ideensammlung / Mind-Map soll beispielsweise ein Textfeld die Eingabe eines Dateinamens erlauben (Abbildung 32). Das Laden einer zuvor gespeicherten Sitzung erfolgt an dieser Stelle durch die Auswahl einer bestimmten Datei aus einer Liste (Abbildung 33). Darüber hinaus soll das Erlernen der im Rahmen der Anwendung umgesetzten Gesten dahingehend unterstützt werden, dass ein Overlay, das die Funktionen der Anwendung sowie die dazu eingesetzten Gesten erklärt und visualisiert, aufgeschaltet werden kann. Die Dimensionen und die Orientierung dieses Overlays wurde im Laufe der Entwicklung des Prototypen geändert, um zu jeder Aktion mehr Text darstellen zu können (siehe Abbildung 34). Schließlich soll ein weiteres Overlay zumindest minimale Informationen bezüglich der aktuell über Bluetooth mit dem Tabletop verbundenen externen Geräte bereitstellen (siehe Abbildung 33). Neben einem Button zum Schließen des Overlays beinhalten alle dieser Elemente ebenso wie die Menüs extra definierte Bereiche zum Bewegen und Rotieren.

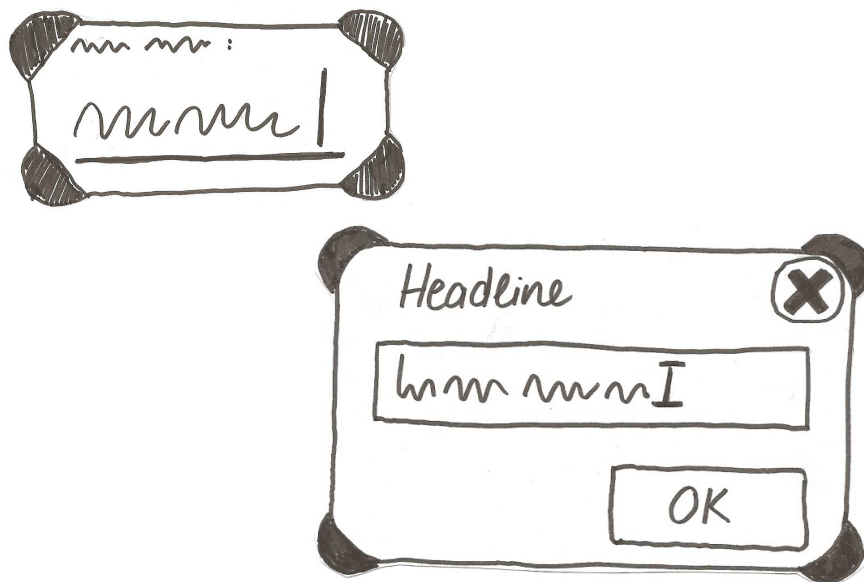


Abbildung 32: Skizzen zur Visualisierung des Speichern-Overlays am Tabletop: Erste Idee, 100x50 mm (oben), finale Skizze 180x130 mm (unten).

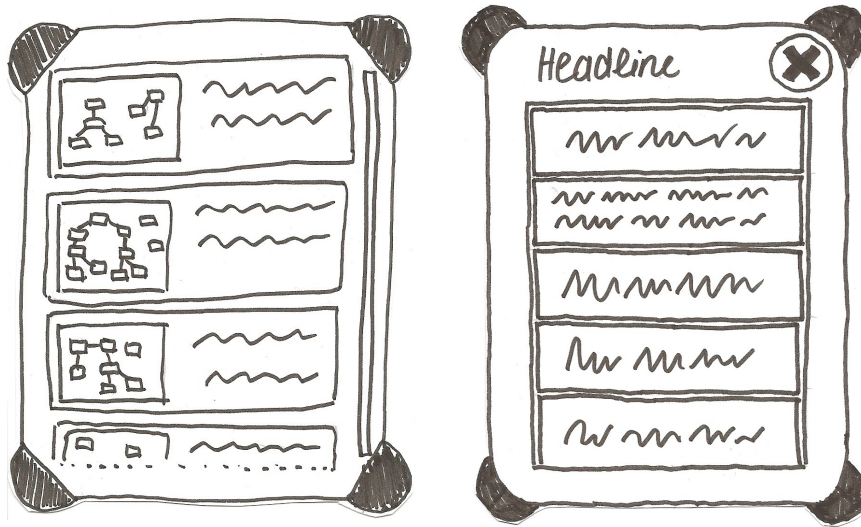


Abbildung 33: Skizzen zur Visualisierung des Laden- bzw. Bluetooth-Overlay (am Tabletop: erste Idee für das Lade-Overlay, 125 x 95 mm (links), finale Skizze Bluetooth/Lade-Overlay, 130 x 100 mm (rechts).

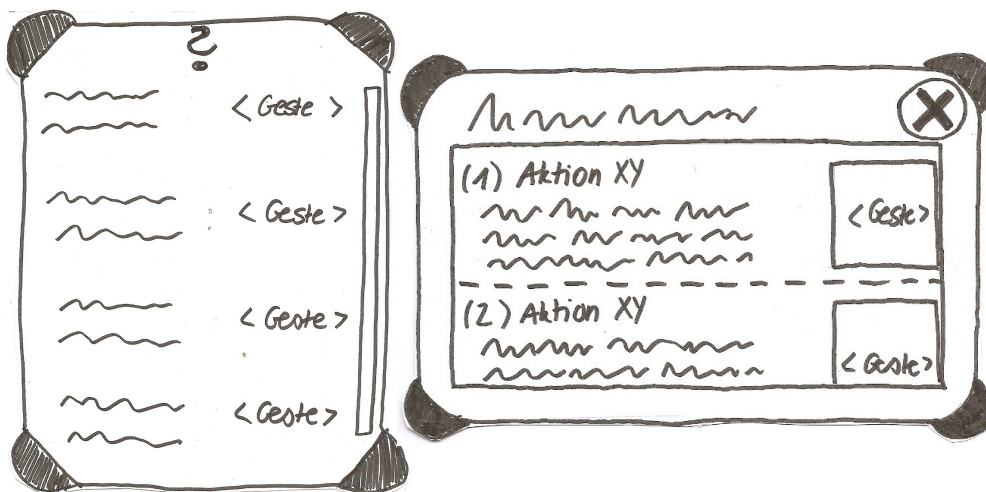


Abbildung 34: Skizzen zur Visualisierung des Hilfe-Overlays am Tabletop: erste Idee, 140 x 110 mm (links) und finale Darstellung, 160 x 100 mm (rechts).

5.5.2 Skizzen Smartphone

Im Gegensatz zur Tabletopschnittstelle lassen sich die Anforderungen, die für die Smartphone-Anwendung im Rahmen des Konzepts definiert wurden, nicht auf einem einzelnen Screen umsetzen. Da sich die Funktionen grob in das Erstellen und Verwalten von Ideen, deren Transfer sowie das Herstellen einer Bluetooth-Verbindung und diesbezüglich relevanter Informationen zum Tabletop einteilen lassen, bietet sich die Umsetzung über das in *Android* bereits verfügbare Tab-Layout-Widget²⁶³ an. Dies stellt für jede Funktionsgruppe einen globalen Reiter mit Icon und Beschreibung zur Verfügung, über das jederzeit zwischen den damit verknüpften *Activities*²⁶⁴ gewechselt werden kann. Die Aufteilung der Benutzerschnittstelle in ‚Ideentransfer‘, ‚Ideenverwaltung‘ und ‚Bluetooth‘ ist auch durch die Tatsache motiviert, dass die Kombination von Ideenerstellung und -transfer (über eine *Drag and Drop*-Geste) in einer *Activity* durch den sehr beschränkten Platz, den mobile Geräte auf ihrem Bildschirm zur Verfügung stellen, kaum praktikabel umsetzbar wäre. Die jeweiligen Inhaltsbereiche der *Activities* bestehen dabei primär aus Listen-Widgets (Abbildung 35).

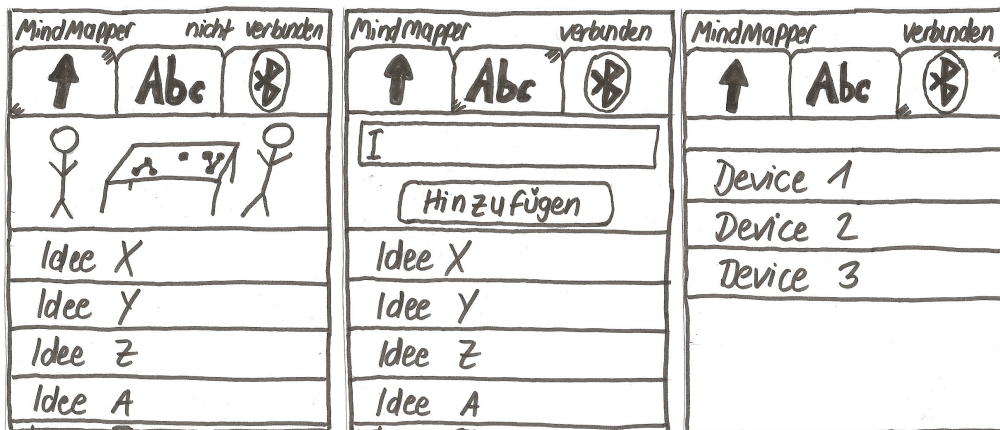


Abbildung 35: Skizzen zur Visualisierung von Ideen für die Benutzerschnittstelle am *Android* Smartphone: Tab-Layout mit *Activity* ‚Transfer‘ (links), *Activity* ‚Ideenverwaltung‘ (mittig) und *Activity* ‚Bluetooth‘ (rechts).

²⁶³ *Android* Tab-Layout: <http://developer.android.com/resources/tutorials/views/hello-tabwidget.html>

²⁶⁴ Eine *Activity* repräsentiert im Rahmen der Entwicklung von Anwendungen für *Android* unter anderem einzelne Anwendungsfenster.

Die *Activity* ‚Ideentransfer‘ umfasst, neben einer Liste der bereits erstellten Ideen, einen Bereich, der den Tabletop visualisieren soll. Das Ziehen und Loslassen eines Listenelements in diesen Bereich resultiert schließlich in dem Transfer der Idee auf den Tabletop, sofern eine Bluetoothverbindung zwischen den beiden Geräten besteht. Die *Activity* ‚Ideenverwaltung‘ ermöglicht die Erstellung einzelner Ideentexte, die durch Bestätigung über einen Button persistent in einer Liste gespeichert werden. Durch das Ausführen der *Tap and Hold*-Geste auf einer bereits erstellten Idee kann diese wieder aus der Liste gelöscht werden. Für die *Activity* ‚Bluetooth‘ ist schließlich in dieser Phase der Konzeption zunächst nur eine Listenübersicht der verfügbaren Bluetooth-Geräte angedacht, die das Verbinden mit einzelnen Geräten ermöglichen soll. Zusätzlich zum Tab-Layout soll darüber hinaus eine Statusleiste Informationen über den aktuellen Bluetooth-Verbindungsstatus bereitstellen.

Durch die Umsetzung von Ideen für die Gestaltung der Benutzerschnittstellen an Tabletop und Smartphone in Form eines Papierprototyp und zusätzlichen Skizzen konnten somit ohne zusätzlichen Entwicklungsaufwand bereits erste Lösungsansätze visualisiert und konkretisiert und darüber hinaus iterativ unter Berücksichtigung der jeweiligen Anforderungen, Möglichkeiten und Grenzen der eingesetzten Hardware und Entwicklungsplattformen weiterentwickelt werden. Dieser technische Kontext der Anwendung soll schließlich im Rahmen des folgenden Kapitels, das die konkrete Umsetzung des zuvor beschriebenen Anwendungskonzeptes erläutert, ebenso kurz skizziert werden.

6 Implementierung

Dieses Kapitel widmet sich der konkreten technischen Umsetzung des zuvor definierten Anwendungskonzepts und den in dessen Rahmen vorgeschlagenen Designlösungen. Neben den für die Ausführung des Systems relevanten technischen Grundlagen und den jeweils verwendeten Entwicklungsumgebungen liegt hier der Fokus vor allem auf der Dokumentierung des strukturellen Aufbaus der Anwendung.

6.1 Technische Grundlagen

Zunächst seien die für die Implementierung relevanten technischen Grundlagen wie die entsprechenden Hardware- und Entwicklungsumgebungen erläutert.

Hardware und Entwicklungsplattform Tabletop

Die Entwicklung des Prototyps für den Tabletop fand am *Evoluce ONE* statt. Dieser interaktive Tisch verfügt über ein berührungsempfindliches LC-Display mit 47 Zoll Bildschirmdiagonale (Abmessungen 1034 mm x 585 mm) mit einer Maximalauflösung von 1920 x 1080 Pixeln (*Full HD*).²⁶⁵ Der Tabletop selbst ist etwa 51 cm hoch und erreicht durch einen zusätzlichen Unterbau (35 cm), der ein komfortables Arbeiten im Stehen ermöglichen soll, eine Höhe von etwa 86 cm. Die Erkennung von Benutzereingaben und getaggten Objekten erfolgt optisch über Infrarotkameras, ähnlich der DI-Technik (siehe Kapitel 3.1). Der *Evoluce ONE* ist mit einem herkömmlichen Desktoprechner mit dem Betriebssystem *Windows 7* verbunden und ermöglicht durch Unterstützung von *Windows 7-Touchgesten* die Interaktion mit üblicher für die Maus- und Tastaturinteraktion entwickelter *Windows*-Software. Darüber hinaus stellt das System eine Kommunikationsschnittstelle in Form des *TUIO*-Protokolls bereit (siehe Kapitel 5.2), welches von dem auch im Rahmen der Arbeit eingesetzten Framework *MT4j* (Version 0.98) unterstützt wird. Laufs, Ruff und Weisbecker beschreiben die Zielsetzung ihres am *Fraunhofer-Institut für Arbeitswirtschaft und Organisation* in Stuttgart (IAO) entwickelten, auf Java basierenden und unter der *GNU GPL*-Lizenz (Version 3)²⁶⁶ veröffentlichten *Open Source*-Multi-Touch-Framework wie folgt:

²⁶⁵ Vgl. *Evoluce* (2010).

²⁶⁶ *GNU General Public License v.3*: <http://www.gnu.org/copyleft/gpl.html>

MT4j is designed for rapid development of graphically rich applications on a variety of common PC hardware and operating systems. The platform has a special focus on reducing existing barriers with regard to multi-touch software development and on making multi-touch software development easier and more efficient.²⁶⁷

Grundlegende Ziele des Frameworks sind somit die Optimierung von Portabilität, der Abstraktion von Eingaben und Performanz sowie der allgemeine Abbau von Barrieren bei der schnellen Entwicklung von Anwendungen mit Multi-Touch-Funktionalität.²⁶⁸ *MT4j* stellt eine Abstraktionsschicht für unterschiedlichste Hardware (Maus, Tastatur, Multi-Touch-Hardware etc.) bereit, welche die Rohdaten in einheitliche Eingabeevents konvertiert. Diese werden schließlich durch eine Verarbeitungsschicht in Form von Gestenevents an die Präsentationsschicht weitergereicht. Die Darstellung in *MT4j* erfolgt in Form von einzelnen Szenen (*Scenes*), die jeweils eine (abstrakte) Arbeitsfläche (*Canvas*) sowie die darin enthaltenen sichtbaren und unsichtbaren (Multi-Touch-) Komponenten (*Components*) umfassen.

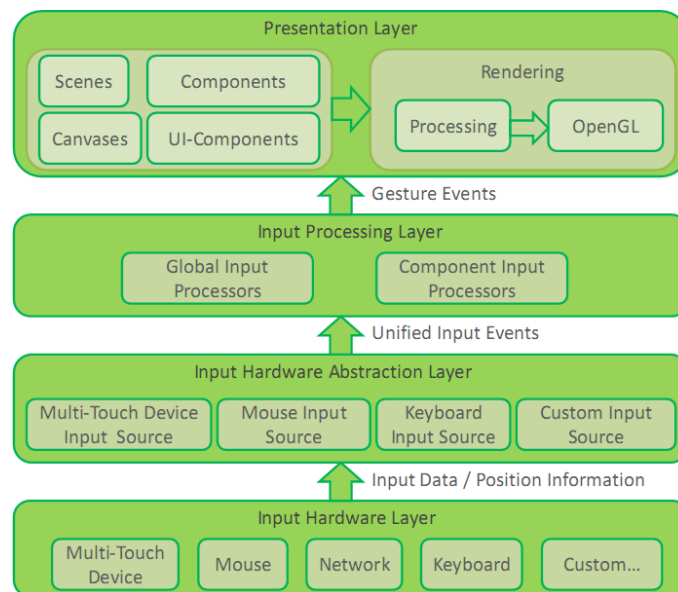


Abbildung 36: Grafische Darstellung der *MT4j*-Architektur (MT4j.org 2011).

²⁶⁷ Laufs, Ruff & Weisbecker (2010, S. 58), Übersetzung durch die Autorin.

²⁶⁸ Vgl. Laufs, Ruff & Zibuschka (2010, S. 2).

Das Rendering der grafischen Ausgabe erfolgt über die in *MT4j* integrierte *Open Source*-Entwicklungsumgebung *Processing*²⁶⁹ sowie die darin verwendeten *OpenGL*-Bibliotheken²⁷⁰ (Abbildung 36).²⁷¹

MT4j stellt bereits einige häufig verwendete Multi-Touch-Gesten (wie *Single* und *Double Tap*, *Drag and Drop* und Gesten zum Rotieren und Skalieren von Objekten) sowie die Erkennung von Unistroke-Gesten mit Hilfe einer Implementierung des *i\$ Unistroke Recognizer*²⁷² zur Verfügung. Letzterer erkennt auf die Tabletopoberfläche gezeichnete Symbole wie Kreise, Rechtecke, Dreiecke, Pfeile etc. sowie Lettern wie X und V. Darüber hinaus bietet *MT4j* bereits einige Multi-Touch-Widgets an, die beliebig erweitert werden können – darunter geometrische Primitive (Rechteck, Ellipse, Linie) sowie komplexere Objekte wie Textfelder, Buttons, scrollbare Listen und virtuelle Tastaturen.

Durch die Entwicklung mit Java sind mit *MT4j* umgesetzte Anwendungen prinzipiell plattformunabhängig.²⁷³ Bezüglich der mit der Anwendung kompatiblen (Erkenner-) Hardware wird lediglich die Unterstützung des *TUIO*-Protokolls und eine *OpenGL*-fähige Grafikkarte vorausgesetzt.

Hardware und Entwicklungsplattform Smartphone

Die Umsetzung der Applikation für Smartphones erfolgte neben dem im *Android SDK* mitgelieferten Emulator primär mit Hilfe eines *LG P500 Optimus One*-Smartphones (3,2 Zoll Bildschirmdiagonale, *Android 2.2 Froyo*). Weitere Tests wurden zusätzlich an einem *Samsung Galaxy S GT-19000* (4 Zoll Bildschirmdiagonale, *Android 2.2 Froyo*) sowie an einem *Samsung Nexus S* (4 Zoll Bildschirmdiagonale, *Android 2.3 Gingerbread*) durchgeführt. Mindestanforderung für das Installieren und Ausführen der Smartphone-Applikation ist *Android 2.1 Eclair* sowie Bluetoothunterstützung.

²⁶⁹ *Processing*: <http://processing.org>

²⁷⁰ *OpenGL*: <http://www.opengl.org/>

²⁷¹ Vgl. Laufs, Ruff & Zibuschka (2010, S. 2 ff.).

²⁷² *i\$ Unistroke Recognizer*: <http://depts.washington.edu/aimgroup/proj/dollar/>

²⁷³ Wegen eines bisher nicht zufriedenstellend lösbaren Problems bezüglich der Minimierung einer *MT4j*-Anwendung muss der *Multi/Touch/Device MindMapper* aufgrund eines entsprechenden Workarounds derzeit noch teilweise auf windowspezifische Funktionalität zurückgreifen (siehe dazu Kapitel 6.3).

Entwicklungsumgebung und Distribution

Die Umsetzung des Programmcodes sowohl für Tabletop als auch für *Android* erfolgte mit der Entwicklungsumgebung *Eclipse* (3.6 *Helios*). Eine effektive und risikofreie Verwaltung der Projektdaten konnte dabei durch den Einsatz des Versionskontrollsystems *Mercurial*²⁷⁴ gewährleistet werden.

Die Distributionsdateien des *Multi/Touch/Device MindMappers* für den Tabletop umfassen etwa 25 MB und stehen, ebenso wie *MT4j*, unter der *GNU GPL*-Lizenz in der Version 3. Die Smartphone-Applikation (95 KB) steht unter der *Apache 2.0*-Lizenz.

Der Quellcode für beide Teile der Anwendung kann online auf *gitHub* unter der URL <http://github.com/SarahW/Multi-Touch-Device-Mind-Mapper> eingesehen und heruntergeladen werden. Unter der URL <http://sarahw.de/mtdmindmapper> finden sich neben den Quelldateien zum aktuellen Stand der Anwendung in gezippter Form zusätzlich bereits kompilierte Versionen des *Multi/Touch/Device MindMappers* (Tabletop: Executable bzw. ausführbare JAR-Datei, Smartphone: APK-Datei) sowie eine Online-Version der Quellcodedokumentation (*JavaDoc*).

²⁷⁴ *Mercurial*: <http://mercurial.selenic.com/>

6.2 Anwendungsstruktur

In der Folge soll schließlich die strukturelle Infrastruktur der *Multi/Touch/Device MindMapper* Anwendung erläutert werden. Diese wurde im Rahmen des Softwareengineeringprozesses zunächst in Form eines allgemeinen Systemmodells definiert, das die Interaktion der einzelnen Hardwarekomponenten untereinander im Anwendungskontext beschreibt (Abbildung 37). Darauf aufbauend wurde ein Architekturmodell jeweils für Tabletop und Smartphone erstellt und erste zentrale Klassen in UML modelliert. Die darauffolgende konkrete Implementierung sowie die dabei umgesetzte Paket- und Klassenstruktur soll schließlich im Fokus dieses Unterkapitels stehen.

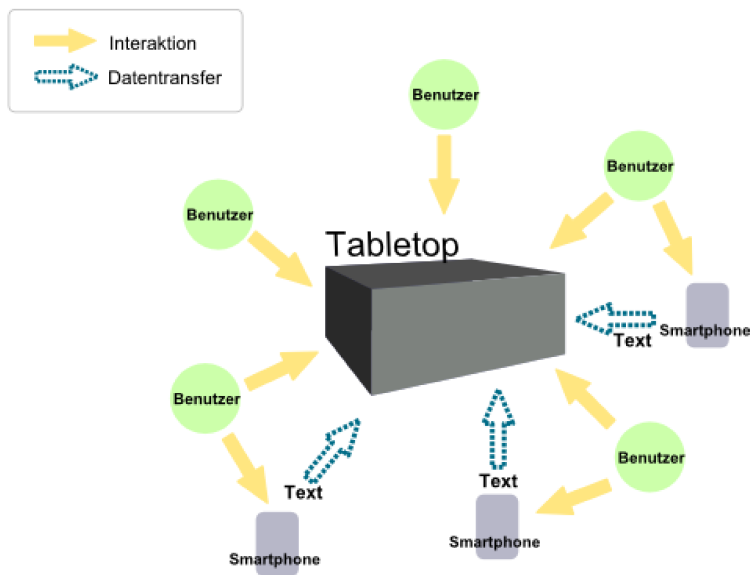


Abbildung 37: Systemmodell der *Multi/Touch/Device MindMapper*-Anwendung.

6.2.1 Architekturmodell

Die im Rahmen der Arbeit umgesetzte Anwendung lässt sich in zwei grundlegende Komponenten unterteilen: Zum einen jener Teil der Anwendung, der auf dem Tabletopsystem läuft (*MT4j*, Java) sowie der Teil, welcher auf zusätzlichen Smartphones (*Android*, Java) ausgeführt wird. Für beide Komponenten wurde versucht, die grundlegende Architektur in Form des *Model-View-Controller-Modells* (MVC) umzusetzen. Dieses Architekturmodell, das sich inzwischen im Bereich der Soft-

warepattern als zentrales Entwicklungsmuster für grafische Benutzerschnittstellen durchgesetzt hat, sieht die Trennung von Datenmodell (*model*), Präsentation (*view*) und Programmsteuerung (*controller*) innerhalb einer Anwendung vor. Ziel dieser Aufteilung „ist ein flexibles Programmdesign, um u.a. eine spätere Änderung oder Erweiterung einfach durchführen zu können und die Wiederverwendbarkeit der einzelnen Komponenten zu ermöglichen“²⁷⁵. Bei *Android* ist dieses Modell bereits in Teilen implizit durch die Auslagerung von Präsentationslogik in XML-Dateien sowie durch den Einsatz von Adapterklassen, welche eine logische Verbindung zwischen Daten und deren konkreter Darstellung herstellen, umgesetzt.

Auf die völlige Separation von Präsentation und Programmsteuerung wurde jedoch im Rahmen der Arbeit verzichtet, da beispielsweise die Verarbeitung von Gesten-Events sowohl bei *MT4j* als auch bei *Android* bereits standardmäßig Teil der Touch-Widgets ist und nicht einfach von deren Präsentation getrennt werden kann. Deshalb kann das für die Anwendung umgesetzte Modell im besten Falle als eine sehr vereinfachte *Model-View-Controller*- bzw. vielmehr als *Model-View*-Architektur betrachtet werden. Dennoch wurde versucht, das Datenmodell von der Präsentationslogik getrennt zu halten. Bei der sehr kompakten *Android*-Anwendung umfasst das Datenmodell im eigentlichen Sinne lediglich die Ideentexte, welche in einer Datenbank gespeichert werden. Die Darstellung dieser Daten erfolgt über die Verbindung von Präsentationskomponenten mit einem Datenbankadapter. Im Rahmen der etwas umfangreicheren Tabletop-Anwendung wurde an dieser Stelle das *Observer*-Pattern zur Benachrichtigung der Präsentationskomponente über Änderungen im Datenmodell eingesetzt.²⁷⁶ Das Datenmodell der Tabletop-Anwendung umfasst dabei die Klassen zur Definition der zu persistierenden Objekte, wie einzelne Ideen und aus Ideen zusammengesetzte Mind-Maps, die mit diesen Objekten möglichen Aktionen (Löschen, Editieren, über Relationen verbinden etc.) sowie die Logik zum Speichern und Laden der aktuellen Sitzung. Somit wird die eigentliche Programmlogik zur Erstellung von Ideen und Mind-Maps von der Darstellung, die durch das *MT4j*-Framework erfolgt, getrennt (siehe auch Kapitel 6.2.2). Darüber hinaus kommuniziert die Programmsteuerung der Präsentation am Tabletop über das *Observer*-Pattern mit dem Bluetooth-Server, um Informationen über die verbundenen Geräte zur Erlangung (Abbildung 38). Die daraus abgeleitete Paket- und Klassenstruktur sei in der Folge schließlich detailliert vorgestellt.

²⁷⁵ Illik (2007, S. 31).

²⁷⁶ Vgl. Eilebrecht & Starke (2010, S. 61 ff.).

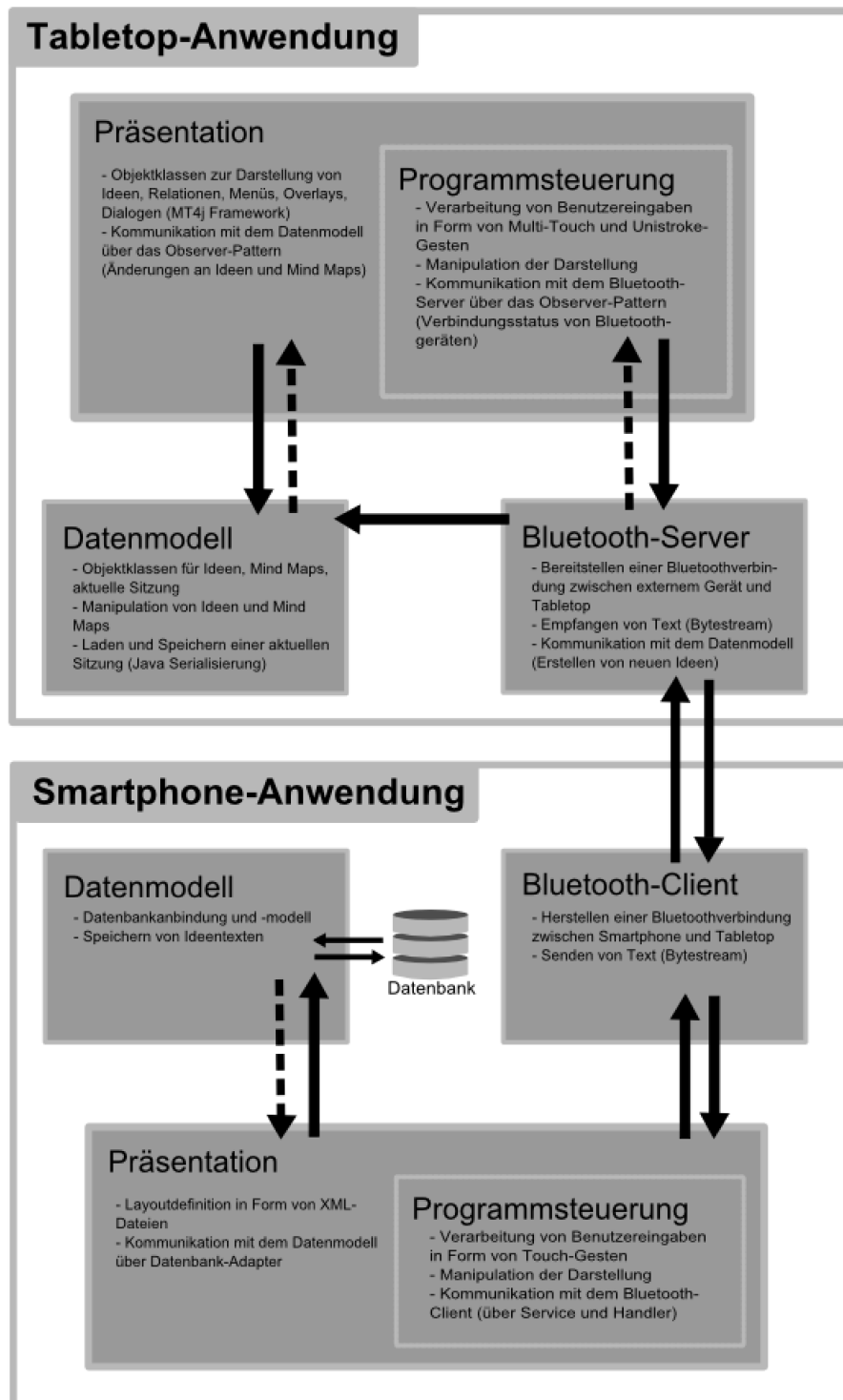


Abbildung 38: Architekturmodell für die *Multi/Touch/Device MindMapper*-Anwendung in Annäherung an das MVC-Pattern.

6.2.2 Paket- und Klassenstruktur

Aus dem allgemeinen Architekturmodell lässt sich eine Paketstruktur für die Anwendung ableiten, welche zusammen mit den für die einzelnen Pakete zentralen Klassen vorgestellt wird. Bezeichnungen von Klassen, die vom *MT4j*-Framework, der Java-API oder anderen Bibliotheken außerhalb der eigentlichen Projektdaten bereitgestellt werden, werden in der Folge zur einfacheren Unterscheidung unterstrichen dargestellt. Methodenparameter werden der Übersichtlichkeit halber durch ‚...‘ ersetzt.

Paket- und Klassenstruktur Tabletop

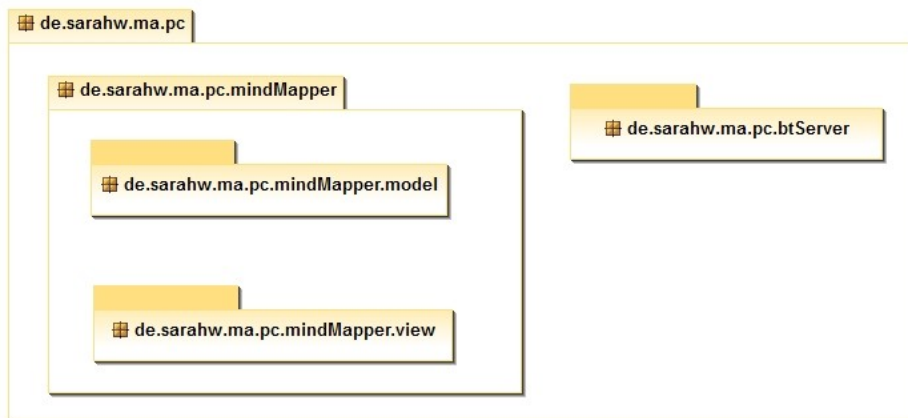


Abbildung 39: Paketstruktur der Tabletop-Anwendung.

Die in Kapitel 6.2.1 erläuterte vereinfachte *Model-View*-Architektur wird im Rahmen des Teils der Anwendung für den Tabletop auf die Paketstruktur abgebildet (Abbildung 39). So lässt sich ein allgemeines Paket `mindMapper` definieren, das neben der `main`-Methode paketübergreifende Werkzeuge bereitstellt. Innerhalb dieses Pakets werden Klassen, welche die benötigten Daten modellieren und die dazugehörige Logik umsetzen, im Paket `model` zusammengefasst. Klassen, die zur Anzeige und Manipulation dieser Daten dienen, werden im Paket `view` gebündelt. Getrennt davon existiert schließlich ein Paket `btServer`, das den Bluetooth-Server zur Verbindung von Smartphones bereitstellt. Anschließend soll eine Auswahl der Klassen dieser Pakete näher beschrieben und anhand der konkreten Implementierung visualisiert werden.²⁷⁷

²⁷⁷ Eine detaillierte Dokumentation aller Klassen sowie deren Eigenschaften und Methoden findet sich im digitalen Anhang auf der der Arbeit beigelegten Daten-CD in Form der aus dem Quellcode generierten *JavaDoc*, ebenfalls verfügbar unter <http://www.sarahw.de/mtdmindmapper>

Paket mindMapper

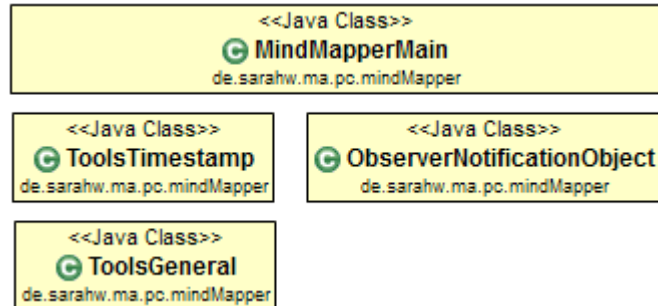


Abbildung 40: Klassenübersicht für das Paket mindMapper.

Das Paket mindMapper enthält mit der Klasse MindMapperMain die main-Methode der Anwendung. Darüber hinaus stellt das Paket mit den Klassen ToolsGeneral und ToolsTimestamp paketübergreifende Werkzeuge sowie mit der Klasse ObserverNotificationObject ein Objekt für die erweiterte Kommunikation zwischen Observer und Observable-Klassen bereit (Abbildung 40).

Paket model

Das Paket model umfasst die Definition aller im Rahmen der Anwendung (persistent) genutzten Daten und zugehöriger Methoden sowie die Logik zur Serialisierung und Deserialisierung dieser Daten. Bei Instanziierung der Klasse AppModel wird eine Instanz der Klasse MindMapCollection (abgeleitet von Observable) erzeugt (Anwendung des *Singleton*-Pattern)²⁷⁸.

MindMapCollection und MindMap

Die Klasse MindMapCollection repräsentiert eine Sammlung aller im Rahmen der Anwendung verfügbaren MindMap-Dateien, das heißt persistent gespeicherten Sitzungen in Form einer Liste. Bei Instanziierung von MindMapCollection wird eine leere MindMap durch das Erzeugen eines Objekts der Klasse MindMap (abgeleitet von Observable, implementiert Serializable) geladen. Die Klasse MindMapCollection verfügt darüber hinaus mit loadMindMap(...) über eine Methode zum Laden einer zuvor serialisierten MindMap über die Klasse MindMap-Deserializer (nicht abgebildet). Ferner kommuniziert sie Änderungen bezüglich einer neu geladenen MindMap an registrierte Observer (z.B. Klasse Mind-

²⁷⁸ Vgl. Eilebrecht & Starke (2010, S. 35 ff.).

MapScene im Paket view). Das jeweils aktuell geladene bzw. deserialisierte Mind-Map-Objekt repräsentiert an dieser Stelle die aktuelle Sitzung bzw. Arbeitsfläche. Eine MindMap besteht wiederum aus einer Liste von 0 bis n einzelnen Ideenknoten (Klasse IdeaNode, abgeleitet von Node<NodeData>) sowie einer Liste 0 bis n Mind-Maps (Klasse Map, abgeleitet von Tree<NodeData>)²⁷⁹, welche die Verbindungen zwischen einzelnen Ideenknoten in Form einer Baumstruktur beschreiben (Abbildung 41). Die Methode saveMindMap() ermöglicht schließlich das Speichern der aktuellen MindMap-Instanz in einer Datei in Form einer Java-Serialisierung über die Klasse MindMapSerializer (nicht abgebildet).

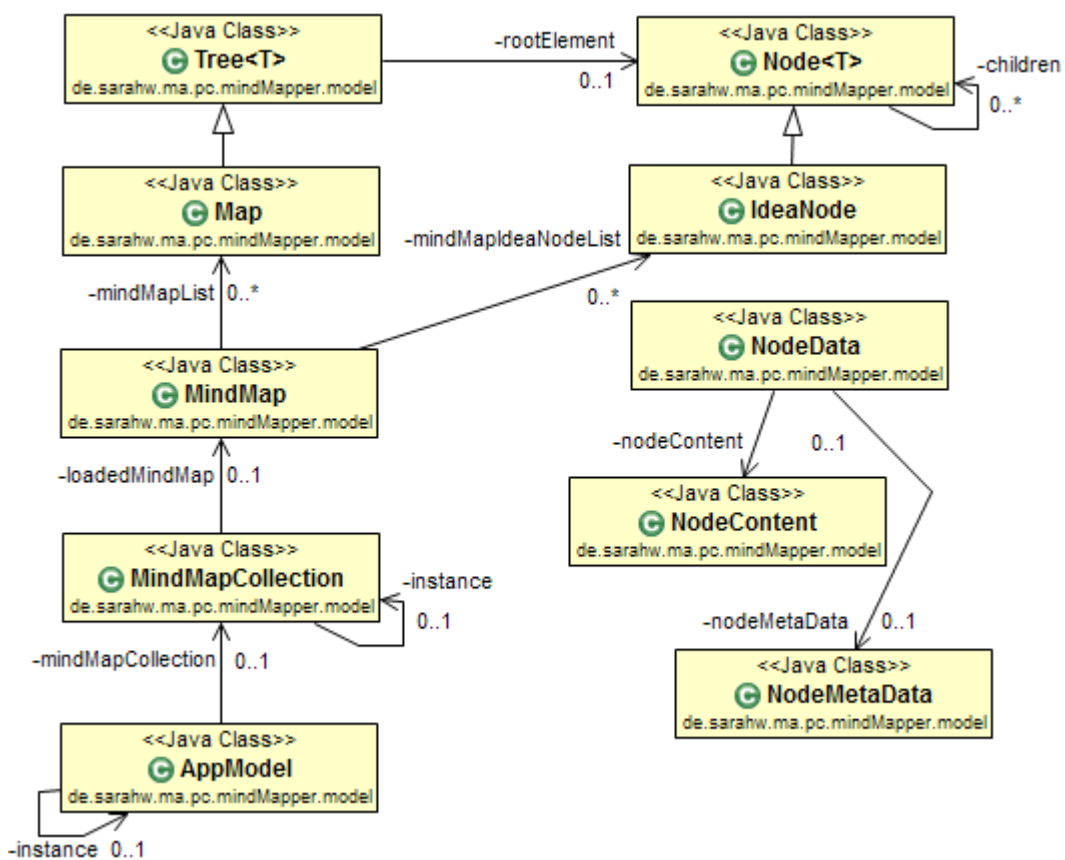


Abbildung 41: Auszug aus dem UML-Klassendiagramm des Pakets model mit Vererbungs- und Kompositionsbeziehungen.

²⁷⁹ Implementierung einer generischen Baumstruktur durch die von Pal (2006) adaptierten generischen Klassen Node<T> und Tree<T>.

IdeaNode und Node<T>

IdeaNode-Objekte als konkrete Implementierung (abgeleitet von *Node<NodeData>*) der generischen Klasse *Node<T>* (abgeleitet von *Observable*, implementiert *Serializable*) verfügen über eine Membervariable vom Typ *NodeData* (implementiert *Serializable*). Die Klasse *NodeData* fungiert hier als Containerklasse sowohl für ein Objekt vom Typ *NodeMetaData* (implementiert *Serializable*), welches unter anderem Eigenschaften, welche die Rotation einer Idee in Grad, deren Position (x- und y-Koordinate) oder dessen Quelle (Tabletop oder externes Gerät) abbilden, enthält als auch für ein Objekt vom Typ *NodeContentData* (implementiert *Serializable*), welches den Inhalt eines Ideenknotens, im Falle der hier umgesetzten Anwendung also den Text einer Idee, enthält. Die Trennung in *NodeMetaData* und *NodeContentData* soll an dieser Stelle die Möglichkeit der flexiblen Weiterentwicklung der Anwendung hin zu alternativen Inhaltstypen über reinen Text hinaus offen halten. Die Elternklasse *Node<T>* definiert zudem für *IdeaNode* eine Membervariable in Form einer Liste von *Node<NodeData>* Objekten, welcher die durch eine Eltern-Kind-Relation verknüpften Kind-*IdeaNodes* hinzugefügt werden. Das Hinzufügen von Kindelementen unterliegt an dieser Stelle diversen Einschränkungen: steht das neu hinzuzufügende Kindelement bereits in Beziehung mit einem Eltern-Ideenknoten oder würde durch die neue Relation das Prinzip der Baumstruktur anderweitig verletzt (beispielsweise durch Rekursion), wird der potentielle Kindknoten der Struktur (und der Map) nicht hinzugefügt.

Veränderungen aller Eigenschaften eines *IdeaNode* Objekts werden schließlich registrierten *Observern* (hier: dem entsprechenden *IdeaNodeView* Objekt aus dem Paket *view*) kommuniziert.

Map und Tree<T>

Die Verknüpfung von einzelnen Ideenknoten in Form von Mind-Maps wird bereits durch die (vererbte) Membervariable *children* der Klasse *IdeaNode* repräsentiert, zur Repräsentation einer ganzen Mind-Map dient jedoch die Klasse *Map* (abgeleitet von *Tree<NodeData>*). Diese Klasse beschreibt eine Baumstruktur durch die Membervariable *rootElement* vom Typ *IdeaNode*, die das Wurzelement der Mind-Map repräsentiert, und in der Eigenschaft *children* weitere Kindelemente definieren kann. Zusätzliche von *Tree<T>* geerbte Methoden wie *walk(...)* und *toList()* ermöglichen die rekursive top-down Traversierung dieser Baumstruktur sowie deren lineare Ausgabe. Obwohl das Konzept einer Map

für die Darstellung von miteinander verbundenen Ideenknoten in Form von `IdeanodeViews` und `RelationViews` (Paket `view`) nicht benötigt wird, da hier bereits die Assoziation zwischen Eltern- und Kindknoten ausreicht, kommt die durch die `Map`-Klasse bereitgestellte Methode `toList()` über die ebenfalls in `Map` enthaltene Methode `containsIdeaNode(...)` zum Einsatz, welche im Rahmen der Überprüfung von Einschränkungen beim Hinzufügen von Kindknoten zu einer `IdeaNode` durch die Methode `addIdeaChild(...)` aufgerufen wird, um rekursive Baumstrukturen zu verhindern.

Paket `btServer`

Die Klassen des Pakets `btServer` stellen einen Bluetooth-Server bereit, der auf Verbindungsanfragen von externen Geräten wartet, bei Eingehen einer neuen Anfrage eine Verbindung zwischen den beiden Geräten herstellt und die von den verbundenen Smartphones transferierten Ideentexte aus einem `ByteStream` ausliest, um schließlich im Datenmodell das Erstellen einer neuen `IdeaNode` anzustoßen. Die Umsetzung dieses Bluetooth-Servers erfolgt mit Hilfe der Java-Bibliothek *BlueCove*²⁸⁰ und stellt größtenteils eine Modifikation des von Luu Gia Thuy entwickelten *Remote Bluetooth Android*-Projekts, dem auch große Teile des Quellcodes für den Bluetooth-Client auf *Android*-Seite entnommen wurden, dar.²⁸¹

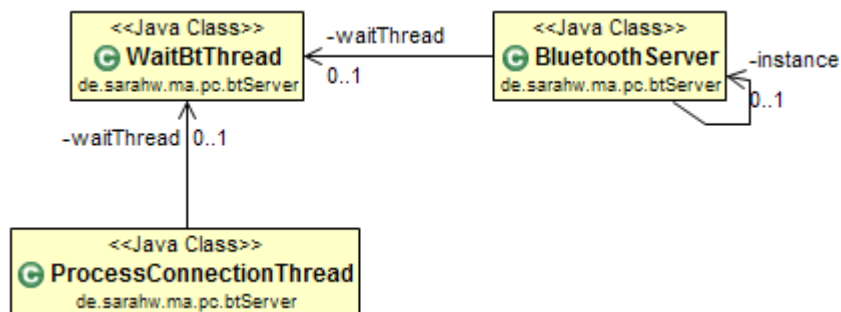


Abbildung 42: Auszug aus dem UML-Klassendiagramm des Pakets `btServer` mit Kompositionsbeziehungen.

Die Klasse `BluetoothServer` startet bei Instanziierung einen Thread der Klasse `WaitBtThread` (abgeleitet von `Observable`, implementiert `Runnable`), welcher zunächst das lokale Bluetooth-Gerät in den Modus *discoverable* versetzt, damit es für externe Geräte bei einem Scan sichtbar ist (Abbildung 42). In der Folge wird

²⁸⁰ *BlueCove*: <http://www.bluecove.org/>

²⁸¹ Vgl. Thuy (2010).

eine `StreamConnection` geöffnet und innerhalb einer Endlosschleife auf eingehende Verbindungsanfragen gewartet. Erfolgt die Verbindung zwischen Smartphone und Tabletop zum ersten Mal, ist zuvor die Kopplung beider Geräte nötig. Hierfür muss am Smartphone und am Tabletop eine Passphrase abgeglichen und bestätigt werden. Am Tabletop erfolgt dies jedoch außerhalb der *Multi/Touch/Device MindMapper*-Anwendung erst nach Auswahl des Bluetooth-Icons in der *Windows*-Statusleiste, wozu die Anwendung minimiert werden muss. Ist schließlich ein erfolgreicher Verbindungsaufbau mit einem externen Gerät erfolgt, wird ein weiterer Thread der Klasse `ProcessConnectionThread` (implementiert `Runnable`) gestartet und die aktualisierte Liste der verbundenen Geräte an registrierte `Observer` kommuniziert (hier: `MindMapScene` im Paket `view`). Innerhalb des für jede Verbindung existierenden `ProcessConnectionThreads` wird ein `InputStream` geöffnet, aus welchem transferierte Ideentexte als UTF-8 Strings ausgelesen werden. Entspricht die Länge dieses Strings der gültigen Länge für den in einem `NodeContent`-Objekt gespeicherten Text, wird über das Modell eine neue `IdeaNode` erstellt.

Paket view

Das Paket `view` umfasst alle Klassen zur Präsentation der Daten sowie der Verarbeitung von Benutzereingaben am Tabletop. Die Klasse `StartMindMapView` (abgeleitet von `MTApplication`) ist für die Initialisierung dieser Komponente zuständig. Es wird zunächst eine neue `MTApplication`-Instanz (*MT4j*-Applikation) mit den in der `Settings.txt` angegebenen Parametern erzeugt. In der Folge wird ein Ladescreen aufgeschaltet, während in einem separaten Thread Bluetooth-Server, Datenmodellkomponenten und schließlich die eigentliche *MT4j*-Szene initialisiert werden. Nach vollständiger Instanziierung der Szenen-Klasse `MindMapScene` wird vom Ladescreen zur eigentlichen Szene gewechselt, in der zunächst nur die beiden Menüs sichtbar sind (Abbildung 43). Darüber hinaus sind zahlreiche eigene (unter Zuhilfenahme der bereits bestehenden Komponentenklassen aus *MT4j* entwickelte) Widgets Teil des Pakets `view`. Für die gewünschte Umsetzung bestimmten Verhaltens, wie beispielsweise die Anforderung, dass Widgets nur bis zu einem gewissen Punkt über den Bildschirmrand hinaus geschoben werden können oder sich Relationen bei Bewegen und Rotieren von Kindelementen flexibel mitbewegen, mussten darüber hinaus die in *MT4j* standardmäßig implementierten Gestenprozessoren und -aktionen erweitert und angepasst werden.

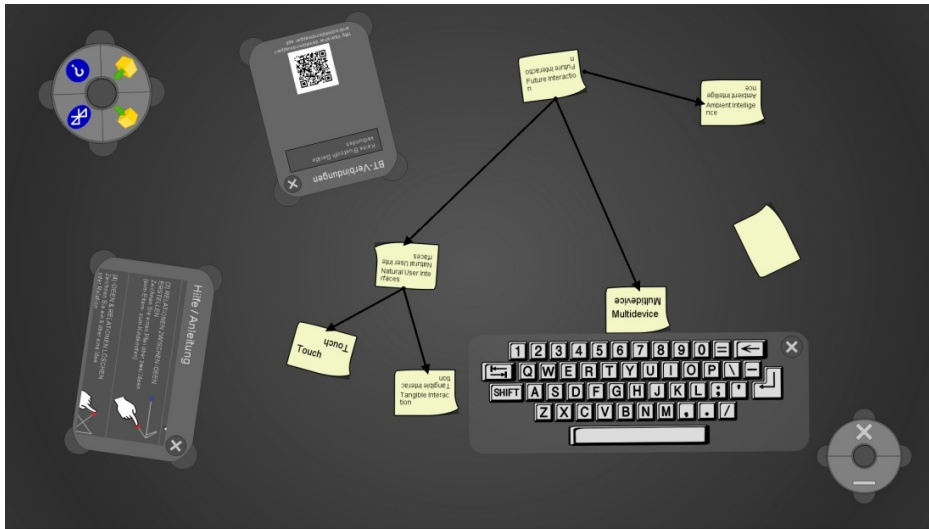


Abbildung 43: Arbeitsfläche der Szene MindMapScene am Tabletop während einer Brainstorming/Mind-Mapping Sitzung.

MindMapScene

Die für das Paket `view` zentrale Klasse `MindMapScene` (abgeleitet von `AbstractScene`, implementiert `Observer`) übernimmt die Initialisierung der Arbeitsfläche sowie die Instanziierung der einzelnen Widgets und stellt Event-Listener für das Erkennen der Unistroke-Gesten sowie weitere zentrale Methoden bereit. Beim Erzeugen der `MindMapScene`-Instanz werden zunächst abhängig von der aktuellen Auflösung Werte für die Größen einzelner Benutzerschnittstellenelemente berechnet. Die Darstellung der einzelnen Widgets erfolgt somit über relative und nicht über absolute Werte, um auch eine passende Darstellung für alternative Auflösungen ermöglichen zu können. Dies wird zudem durch den ausschließlichen Einsatz von Vektorgrafiken (SVG-Format) für die einzelnen Widgets unterstützt.²⁸²

Darüber hinaus wird die Arbeitsfläche der Szene in vier Bereiche aufgeteilt, um die automatische Ausrichtung von neuen Ideenknoten sowie von über das Hauptmenü aufgeschalteten Overlays zu bestimmen. Die durch die mit dem Tabletop verbundenen Smartphones erstellten Ideen erscheinen nach dem Transfer mit einer Animation in der Mitte der Arbeitsfläche (Abbildung 44).

²⁸² Bisher konnte keine Lösung für die Anpassung von Schriften abhängig von der Auflösung umgesetzt werden, weshalb die Anwendung primär für die Auflösung 1920 x 1080 Pixel optimiert ist. Widgets und ihre einzelnen Komponenten werden somit zwar abhängig von der Auflösung (nach dem Start) skaliert, Schriftgrößen können jedoch derzeit noch nicht angepasst werden.

Das Hinzufügen eines Gesten-Listeners zum Canvas der MindMapScene ermöglicht schließlich die Erkennung der Unistroke-Gesten ‚Rechteck/Kreis‘ (Erstellung einer neuen Idee), ‚Pfeil‘ (Verbinden zweier Ideen) und ‚X‘ (Löschen einer Idee oder einer Relation) auf der Oberfläche (Abbildung 45). Alle Berührungseingaben, die direkt auf der Arbeitsfläche und nicht auf einem Benutzerschnittstellenelement erfolgen, werden dabei als Unistroke-Geste erfasst.

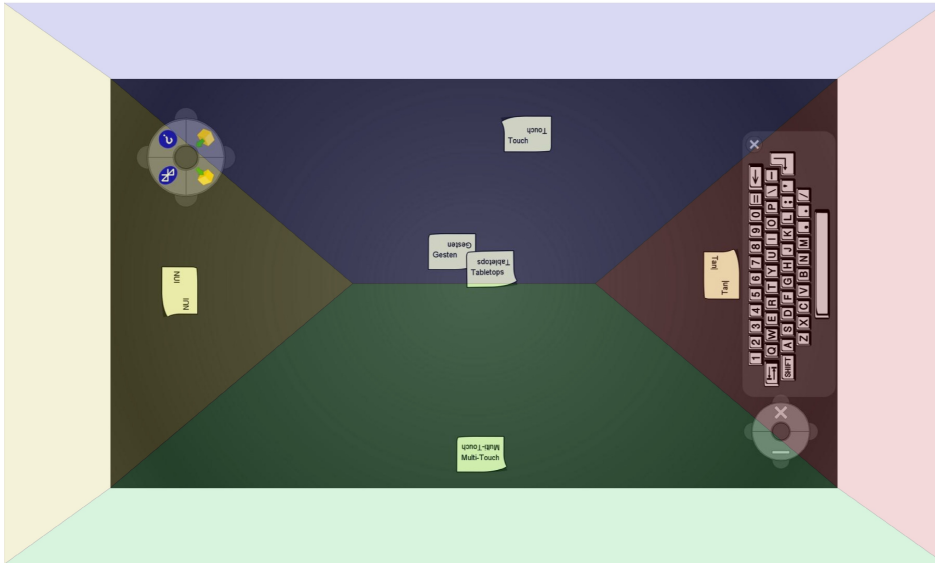


Abbildung 44: Aufteilung der Tabletopoberfläche in vier Bereiche zur automatischen Ausrichtung von neuen Ideen und Menü-Overlays.

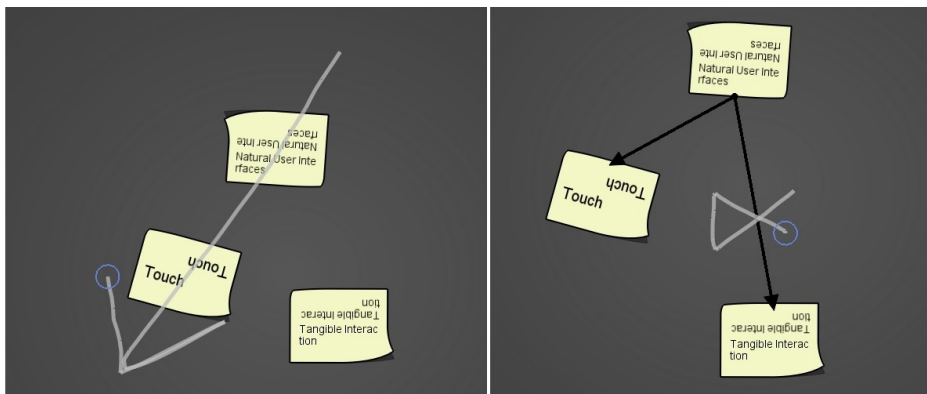


Abbildung 45: Unistroke-Gesten zur Erstellung von Relationen (Pfeilgeste) und Löschen von Ideen und Relationen (X-Geste).

Je nach Erfolg der Erkennung einer solchen Geste und der Möglichkeit einer Aktion an dieser Stelle der Arbeitsoberfläche erfolgt eine Rückmeldung vom System durch Einfärben der gezeichneten Unistroke-Geste (Abbildung 46). Wird die Geste nicht erkannt, wird sie in rot dargestellt. Wird sie erkannt und kann an dieser Stelle die damit verknüpfte Aktion auch ausgeführt werden, wird sie grün dargestellt. Die gelbe Darstellung der Unistroke-Geste dient schließlich zur Visualisierung des Falles, dass zwar die Geste selbst korrekt erkannt wurde, die Aktion an dieser Stelle jedoch nicht möglich ist, beispielsweise das Verknüpfen zweier Ideenknoten aufgrund bestimmter Einschränkungen, die sich durch die Baumstruktur ergeben.

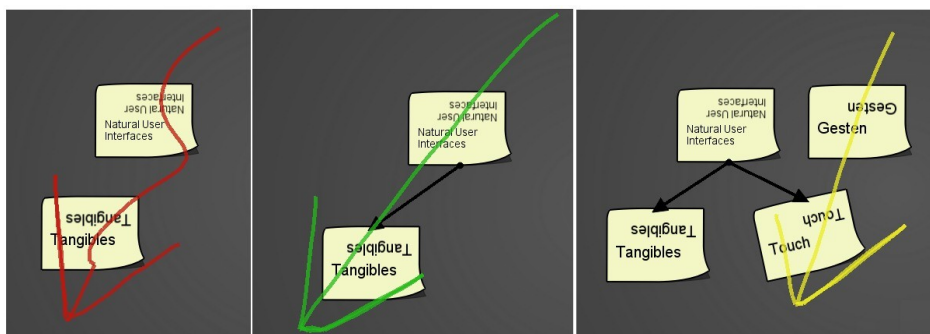


Abbildung 46: Feedback bei der Erkennung von Unistroke-Gesten: a) Rot – Geste nicht erkannt, b) Grün – Geste erkannt, Aktion erfolgt c) Gelb - Geste erkannt, jedoch keine Aktion möglich (hier: bereits Elternknoten vorhanden).

Schließlich wird das Szenobjekt vom Typ `MindMapScene` als `Observer` bei der `MindMapCollection`-Instanz, der aktuell geladenen `MindMap` und der Instanz des `WaitBtThreads` registriert, um Änderungen im Datenmodell und bezüglich Bluetooth-Verbindungen kommuniziert zu bekommen.

Weitere Kommunikation mit dem Datenmodell erfolgt über diverse Methodenaufrufe. Die Interaktion zwischen Präsentation und Datenmodell soll an dieser Stelle anhand der Erstellung einer neuen Idee am interaktiven Tisch exemplarisch vorgestellt werden: Wenn eine Rechteck-Geste auf der Oberfläche erkannt wurde, wird über die Methode `createModelIdeaNode(...)` eine neue `IdeaNode` für die aktuell geladene `MindMap`-Instanz im Datenmodell erstellt. Sofern diese Änderung des Datenmodells erfolgreich durchgeführt werden konnte, wird diese Zustandsänderung der `MindMapScene`-Instanz als registriertem `Observer` erneut kommuniziert, woraufhin erst über `createIdeaNodeView(...)` eine entsprechende `View`-Instanz der Klasse `IdeaNodeView` für die neue `IdeaNode` erzeugt und angezeigt wird. Ähnlich erfolgt die Kommunikation bei anderen Aktionen

am Tabletop, wie dem Löschen von Ideen und Relationen oder dem Verknüpfen von einzelnen Ideen – zunächst erfolgen die Änderungen nach Prüfen bestimmter Bedingungen immer erst im Datenmodell, welches diese Änderungen den jeweiligen `Observern` kommuniziert, woraufhin die entsprechenden Anpassungen in der Präsentation vorgenommen werden. Einzige Ausnahme bilden die Eigenschaften Text, Rotation und Position eines `IdeaNodeView` bzw. einer `IdeaNode`, welche aufgrund der unmittelbar sichtbaren und sehr häufigen Änderungen durch die Präsentationskomponente immer erst beim Speichern einer aktuellen Sitzung im Datenmodell synchronisiert werden.

IdeaNodeView und RelationView

Die Darstellung einzelner `IdeaNodes` erfolgt über die Klasse `IdeaNodeView` (abgeleitet von `MTRectangle`, implementiert `Observer`). Visualisiert wird eine Idee über einen digitalen Notizzettel mit einem an der horizontalen Achse gespiegelten, zweizeiligen Textfeld (Abbildung 46). Besonderheit des Textfeldes vom Typ `MTTextFieldVarLinesFontSwitch` (abgeleitet von `MTTextFieldVarLines`) im Vergleich zu dem standardmäßig durch `MT4j` zur Verfügung gestellten einzeiligen Textfeld vom Typ `MTTextField` ist der automatische Zeilenumbruch und das automatische Verringern der Schriftgröße ab einer gewissen Textmenge zur optimalen Ausnutzung des verfügbaren Platzes.²⁸⁵

Gemäß des hierarchischen Aufbaus von Komponenten in `MT4j` werden weitere Elemente, wie ein `NodeContentContainer`, der wiederum das Textfeld vom Typ `MTTextFieldVarLinesFontSwitch` enthält, dem `IdeaNodeView` über die vererbte Methode `addChild(...)` als Kindelemente hinzugefügt (Abbildung 47). Ein weiteres Kindelement von `IdeaNodeView` ist ein `KeyboardContainer`, der ein `CustomKeyboard` (abgeleitet von `MForKeyboard`) beinhaltet. Dieses virtuelle Keyboard wird aufgeschaltet, wenn eine Idee gerade neu erstellt wurde oder der Benutzer zweimal kurz hintereinander auf einen `IdeaNodeView` tippt (*Double Tap*) und erlaubt das Editieren des Textes der Idee. Der `KeyboardContainer` verhindert hier das Weiterleiten von Gestenevents in der Hierarchie nach oben, so dass das Bewegen und Rotieren des Keyboards keinen Einfluss auf Position und Ausrichtung des Ideenknotens hat. Ein weiteres Kindelement der `IdeaNodeView`-Komponente ist schließlich ein `MarkerContainer` (abgeleitet von `MComponent`), der die vier Punkte zwischen den Eckpunkten eines `IdeaNodeView` definiert, an

²⁸⁵ Bisher nicht implementiert ist eine automatische Wort- und Silbentrennung, das heißt, ist die maximale Breite pro Zeile erreicht, wird nach dem Zeilenumbruch lediglich das nächste Zeichen in die folgende Zeile versetzt.

die ein `RelationView` (abgeleitet von `MTLine`) – welcher die Eltern-Kind-Relation zwischen zwei miteinander verknüpften `IdeaNodeViews` mit einem schwarzen Pfeil visualisiert – andocken kann (Abbildung 47).

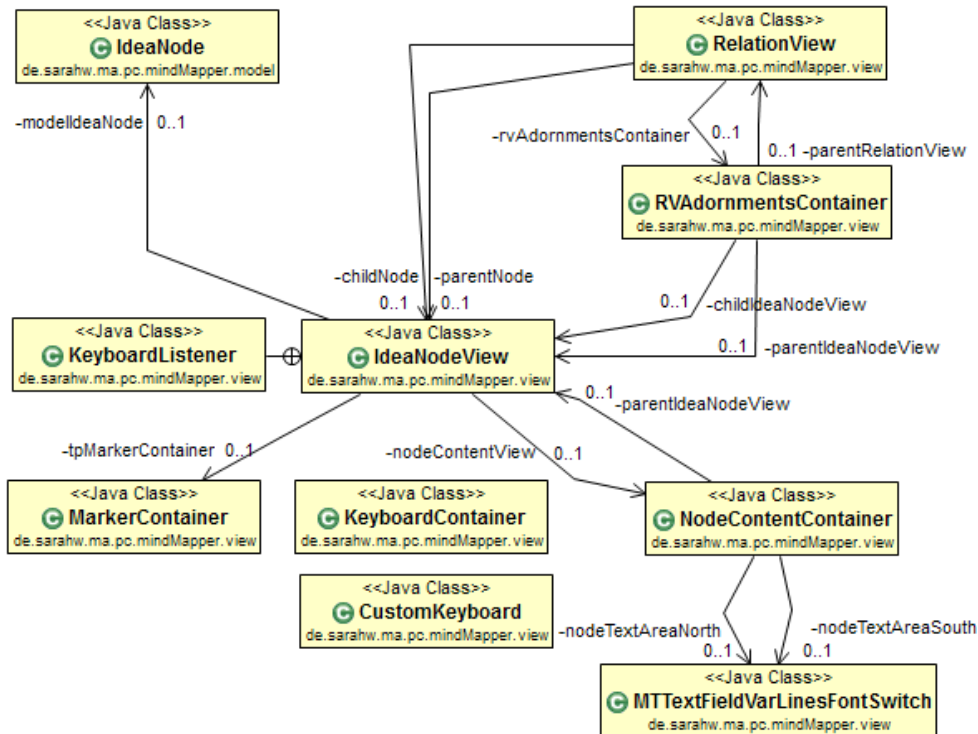


Abbildung 47: Auszug aus dem UML-Klassendiagramm des Pakets `view` um die Klasse `IdeaNodeView`.

Über den `Observer`-Status wird ein `IdeaNodeView`-Objekt über Änderungen des verknüpften `IdeaNode`-Objekts im Datenmodell informiert und kann so die Verknüpfungen zwischen `IdeaNodes` überwachen, um die Präsentation mittels der Anzeige und Entfernen von `RelationViews` zu synchronisieren.

`IdeaNodeViews` können (durch Ableitung von der `MT4j`-Klasse `MTRectangle`) mittels Touch-Gesten bewegt und rotiert sowie zusätzlich mit einer Flick-Geste über die Arbeitsfläche geworfen werden. Dasselbe gilt für `RelationViews`, wobei die jeweiligen Aktionen gleichsam Auswirkungen auf alle damit verknüpften `RelationViews` und `IdeaNodeViews` haben. Das Bewegen, Rotieren und Werfen eines Wurzel-Ideenknotens einer aus `IdeaNodeViews` und `RelationViews` aufgebauten Mind-Map bewegt, rotiert und wirft ebenso die gesamte Mind-Map. Wird ein `IdeaNodeView` tiefer in der Hierarchie der Mind-Map bewegt oder rotiert (Werfen ist hier nicht möglich), werden vorhandene Kind-`IdeaNodeViews` und

RelationViews gleichsam transformiert; der Eltern-Ideenknoten und mit diesem verknüpfte IdeaNodeViews und RelationViews werden dadurch jedoch nicht verändert, „von unten nach oben“ wirkt sich die Manipulation lediglich auf den mit dem Eltern-Ideenknoten verknüpften RelationView aus (Veränderung der Ausrichtung, Länge und gegebenenfalls der Ankerpunkte des Pfeils).

MainMenu und WindowMenu

MT4j stellt mit den Klassen *MTSquareMenu* und *MTHexagonMenu* lediglich Menüs mit einer primären Ausrichtung bereit – das für die im Rahmen dieser Arbeit umgesetzte Anwendung konzipierte kreisförmige Menü musste somit ebenfalls eigens implementiert werden. Dabei wurde durch den Einsatz abstrakter Klassen insbesondere Wert auf die Erweiterbarkeit und Wiederverwendbarkeit dieses Widgets gelegt. So definiert die Klasse *AbstractCircularMenu* (abgeleitet von *MTEllipse*) die grundlegenden Eigenschaften für ein kreisförmiges Menü mit maximal acht Buttons und *Menu Handles* (Griffen) (Abbildung 48).

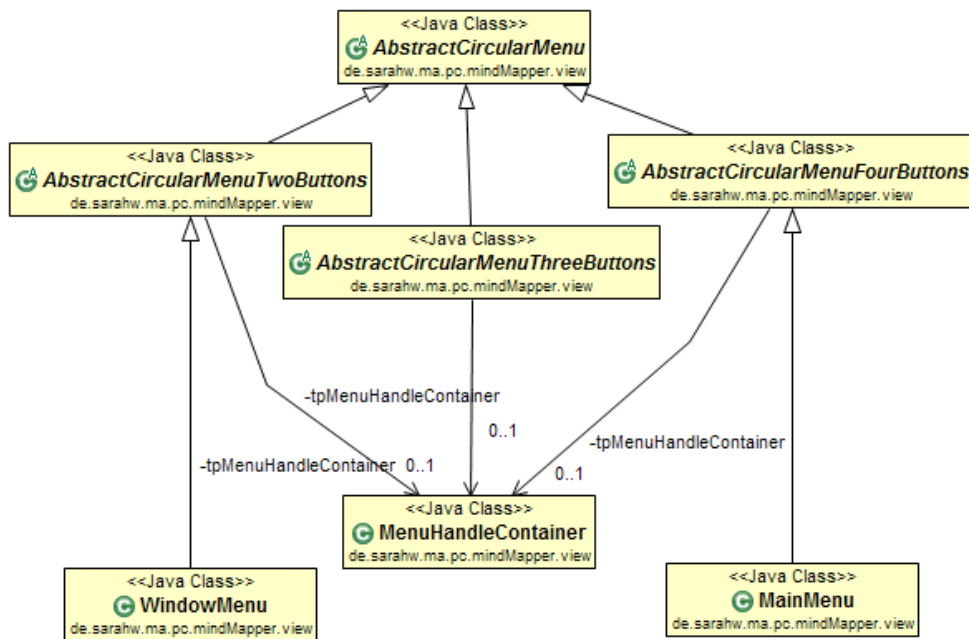


Abbildung 48: Auszug aus dem UML-Klassendiagramm des Pakets view um die Klassen WindowMenu und MainMenu.

Davon abgeleitete Klassen wie *AbstractCircularTwoButtons* definieren darüber hinaus die exakte Anzahl und Positionierung der einzelnen Buttons (vom

Typ `MTSvgButton`) und der Griffe am Rand des Menüs durch einen `MenuHandleContainer` (abgeleitet von `MarkerContainer`). Durch die Implementierung dieser abstrakten Klassen in Form von einzelnen Menüs (hier: `WindowMenu` und `MainMenu`) können schließlich die individuellen Aktionen, die bei der Auswahl einzelner Menüpunkte bzw. Buttons ausgeführt werden, spezifiziert werden (Abbildung 48). Gemäß der während der Konzeptionsphase abgeleiteten Anforderungen und den ersten mit Hilfe des Prototyps umgesetzten Designideen erfolgt die Aufteilung der über die Menüs repräsentierten Funktionen in zwei getrennte Menüs sowie die Kennzeichnung einzelner Menüpunkte über Icons (Abbildung 49). Das `MainMenu` enthält vier Buttons, die das Speichern und Laden einer Sitzung sowie das Anzeigen von Informationen zur Verbindung über Bluetooth und einer Übersicht über die im System genutzten Unistroke-Gesten ermöglicht. Das `WindowMenu` enthält zwei Buttons zum Schließen und Minimieren der Anwendung.²⁸⁴

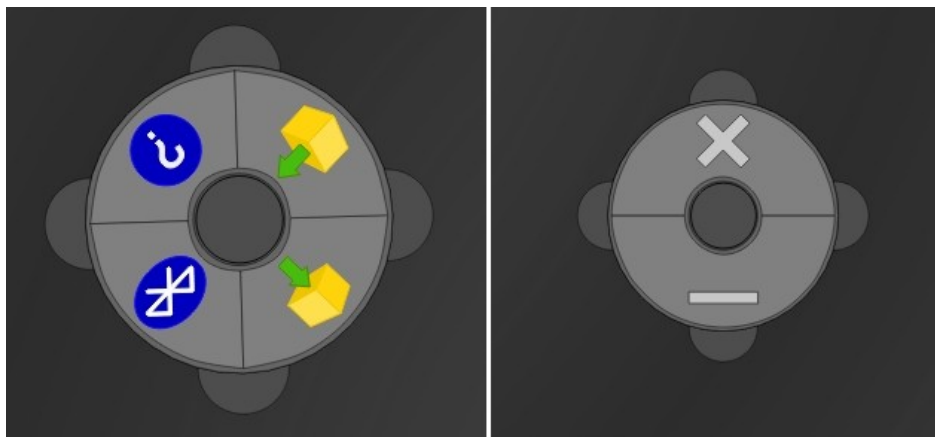


Abbildung 49: Kreisförmige Menüs für grundlegende Funktionen (links) und Fensteraktionen (rechts).

Vor dem Ausführen der Fenster-Aktionen wird jedoch zunächst ein Dialog aufgeschaltet, in dem der Benutzer die gewählte Aktion bestätigen muss oder auch abbrechen kann, um ein ungewolltes und den Arbeitsprozess störendes Schließen oder Minimieren der Anwendung zu verhindern.

Durch das Erben der Methoden und Eigenschaften der `MT4j`-Frameworkklasse `MTEllipse` lassen sich die Menüs ebenfalls frei bewegen und rotieren. Hierfür

²⁸⁴ Die Funktion ‚Verkleinern/Maximieren‘, die noch Teil des Papierprototypen ist, ließ sich im Rahmen des `MT4j`-Frameworks nicht umsetzen und musste deshalb gestrichen werden. Hierzu und zu weiteren Anpassungen des Anwendungskonzept während der Implementierung siehe Kapitel 6.3.

sollen primär die Bereiche in der Mitte und die Griffe an den Rändern dienen. Durch die Art, wie *MT4j* Gesten verarbeitet, ist es jedoch auch möglich, die Menüs über die Buttons zu bewegen und zu rotieren, sofern dadurch das Absetzen der Finger an einer anderen Stelle auf der Oberfläche passiert. Die Auswahl der Menüpunkte erfolgt hier schließlich durch einen *Single Tap*, die Buttons vom Typ `MTSvgButton` signalisieren eine registrierte Berührung dabei durch Verkleinern der Buttonfläche.

Overlays

Zur Anzeige von zusätzlichen Informationen und der Eingabe von Daten werden in der Anwendung Overlays eingesetzt. So wird durch die Auswahl jeder der vier Menüpunkte im MainMenu ein eigenes Overlay aufgeschaltet. Die Position des Overlays entspricht dabei der Position des Menüs; die Ausrichtung erfolgt je nach Zugehörigkeit zu einem der vier definierten Bereiche auf der Arbeitsfläche (Abbildung 44). Alle dieser Overlays sind indirekt von der Klasse `AbstractOverlay` (abgeleitet von `MTRectangle`) abgeleitet, welche die grundlegende Gestenverarbeitung definiert (Abbildung 50).

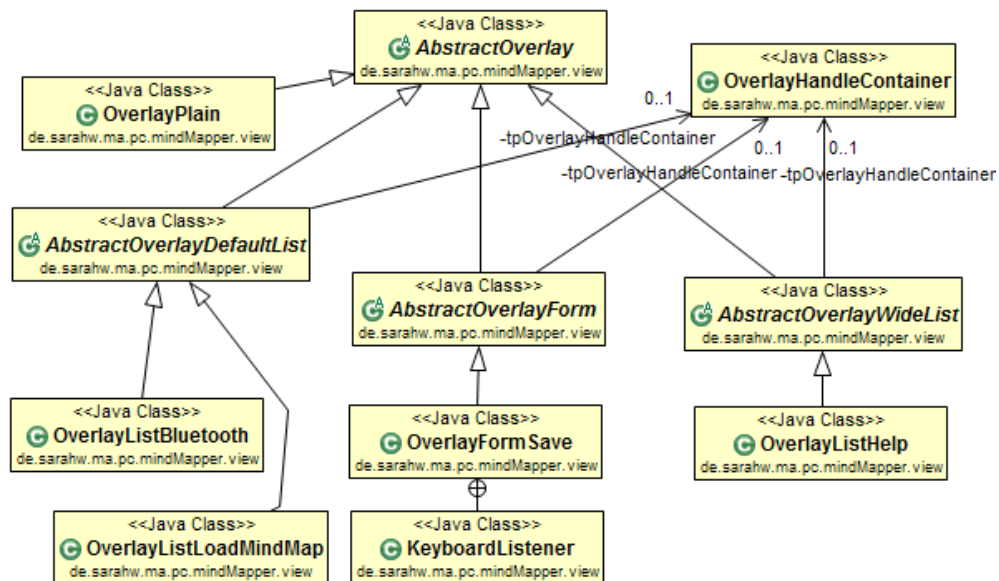


Abbildung 50: Auszug aus dem UML-Klassendiagramm des Pakets view um die Overlay-Klassen.

Weitere abstrakte Subklassen wie `AbstractOverlayForm` oder `AbstractOverlayDefaultList` definieren unterschiedliche Ausprägungen von Overlays, so wie deren Breite und Höhe und die Position und Größe der allen Overlays gemeinsamen Elemente, wie einem Textfeld für die Überschrift und einem Schließen-Button. Ebenso beinhalten alle Overlays die von den Menüs bereits bekannten Griffe, die die Möglichkeit der Rotation und des Bewegens dieser Widgets symbolisieren und umsetzen. Durch die Implementierung der abstrakten Klassen durch Klassen, welche die eigentlichen Overlays repräsentieren, werden die Overlaygerüste schließlich mit Inhalten gefüllt. Die Option ‚Speichern‘ öffnet beispielsweise eine Overlayinstanz der Klasse `OverlayFormSave` (abgeleitet von `AbstractOverlayForm`), die ein Textfeld zur Eingabe eines Dateinamens für die Speicherdatei, einen Button zur Bestätigung der Eingabe und ein zusätzliches Textfeld links neben dem Button zur Anzeige von Informationen bei Ungültigkeit des eingegebenen Namens (zu lang oder Sonderzeichen) enthält. Die Eingabe des Textes erfolgt wie bereits bei `IdeaNodeView` über ein `CustomKeyboard`. Dieses kann ebenfalls über einen Schließen-Button geschlossen und durch einen *Double Tap* auf das Textfeld im Overlay erneut geöffnet werden (Abbildung 51).

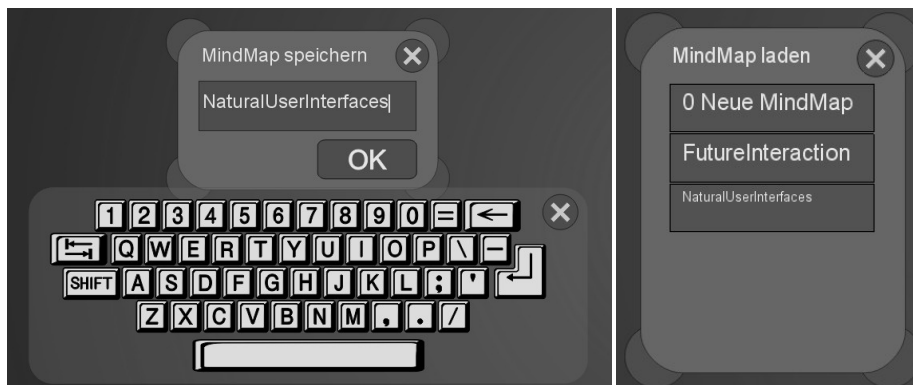


Abbildung 51: Overlays zum Speichern der aktuellen Arbeitsoberfläche (links) und zum Laden einer zuvor gespeicherter Ideensammlung (rechts).

Das Bestätigen über den OK-Button des Speichern-Overlays stößt die Serialisierung des aktuell geladenen `MindMap`-Objekts über das Datenmodell an und zeigt für die (in der Regel sehr kurze) Dauer des Speichervorgangs ein kleines Informations-Overlay vom Typ `OverlayPlain` auf dem Bildschirm an. Während des Speicherns werden die Arbeitsfläche und die darauf befindlichen Elemente für alle Benutzereingaben gesperrt. Je nach Erfolg der Serialisierung wird schließlich

ein Informationsdialog über den Speichererfolg angezeigt. Nach dem Schließen dieses Dialogs wird die Sperrung der Arbeitsfläche wieder aufgehoben.

Der Menüpunkt ‚Laden‘ öffnet schließlich ein Overlay vom Typ `OverlayListLoadMindMap`, in dem eine Liste vom Typ `MtList` angezeigt wird, deren Einträge die zuvor gespeicherten Sitzungen repräsentieren (Abbildung 51). Über eine Wisch-Geste kann in der Liste gescrollt werden, die Auswahl eines Element erfolgt durch einen *Single Tap*. Visuelle Rückmeldung über die getroffene Auswahl erfolgt durch das Ändern der Farbe des Listeneintrags. Wie bereits beim Speichern resultiert das Anstoßen des Ladeprozesses in dem Sperren der Benutzeroberfläche sowie in dem nachfolgenden Aufschalten eines Informationsdialogs über den Erfolg der Aktion.

Die Klasse `OverlayListHelp` definiert darüber hinaus ein Overlay, in dem in einer `MtList` die möglichen Aktionen sowie die dafür eingesetzten Gesten erklärt und visualisiert werden, und das über den durch ein Fragezeichen visualisierten Menüpunkt ‚Hilfe‘ geöffnet werden kann (Abbildung 52). Eine Übersicht über alle Hilfetexte und -visualisierungen findet sich hierzu in Abbildung 53. Eine Besonderheit dieses Overlays im Vergleich zu den anderen Menü-Overlays ist, dass hiervon mehrere Instanzen geöffnet werden können, da anzunehmen ist, dass eventuell auch mehrere Benutzer gleichzeitig die Anleitung konsultieren wollen.

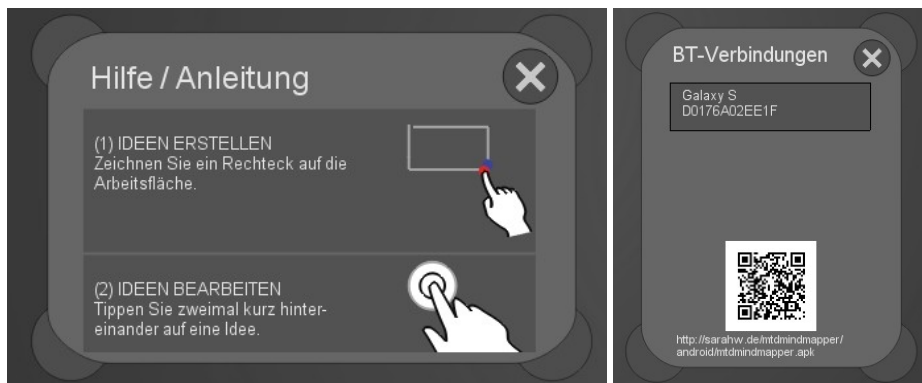


Abbildung 52: Hilfe-Overlay mit Erläuterung und grafischer Darstellung der möglichen Gesten (links, Originalgrafiken bereitgestellt von gestureworks.com) sowie Bluetooth-Overlay zur Anzeige der aktuellen Bluetoothverbindungen und eines Links zur Smartphone-Applikation als QR-Code und Text (rechts).

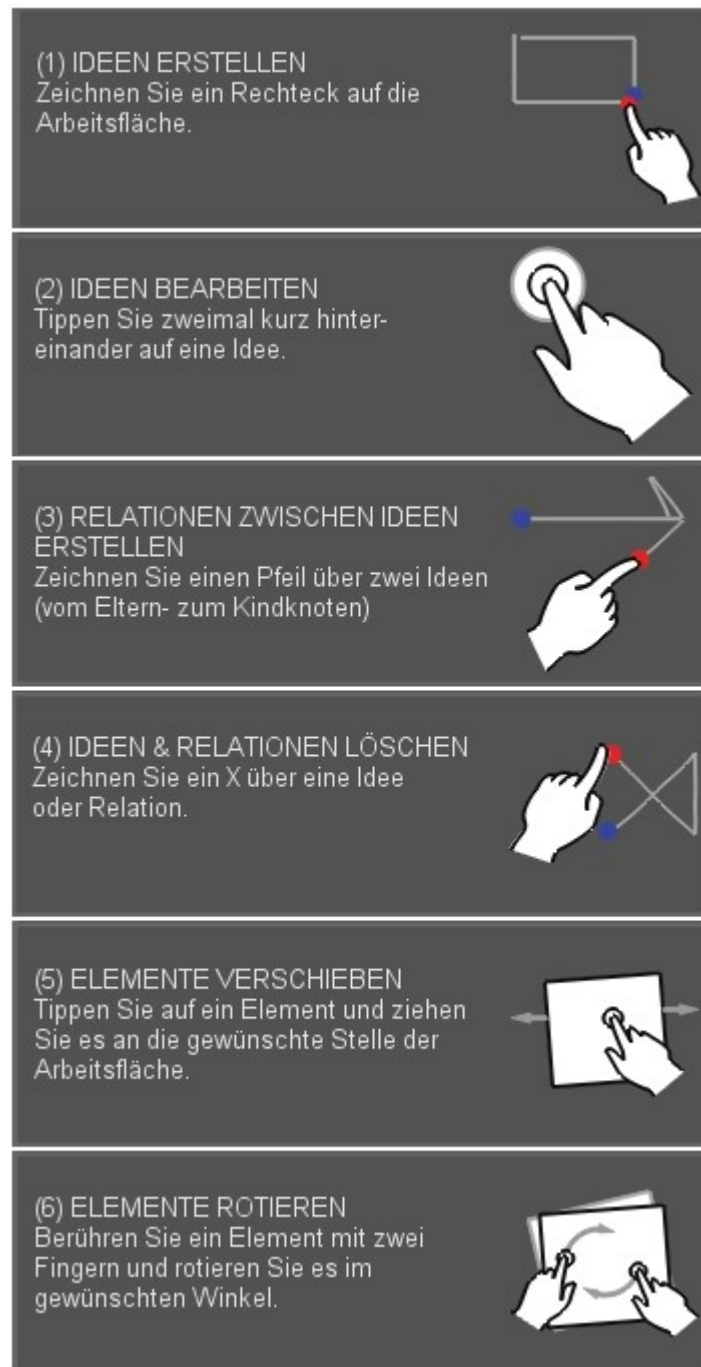


Abbildung 53: Erläuterung der möglichen Aktionen im Hilfe-Overlay mit zusätzlicher grafischer Darstellung der einzelnen Gesten (Originalgrafiken bereitgestellt von gestureworks.com).

Die Menüoption ‚Bluetooth‘ öffnet schließlich ein Overlay der Klasse `OverlayListBluetooth`. Bestandteile dieses Overlays sind eine `MtList`, in der alle aktuell mit dem Tabletop über Bluetooth verbundenen Geräte angezeigt werden, sowie die URL zur Smartphone-Applikation des *Multi/Touch/Device MindMappers* in Form eines *Quick Response* (QR)-Codes, welcher durch Abfotografieren mit dem Smartphone und entsprechender Decodierungs-Applikation in einen Textlink umgewandelt werden kann (Abbildung 52). Für den Fall, dass der Link nicht korrekt erkannt werden kann, ist er zusätzlich in textueller Form angegeben.

Dialoge

Um zusätzliches Feedback über den Erfolg von Aktionen anzuzeigen und die Benutzer vor dem Ausführen von kritischen Aktionen erneut um Bestätigung zu bitten, werden, wie bereits angedeutet, Dialoge verwendet. Diese bestehen letztlich aus einem semi-transparenten Overlay, das in der Mitte der Anwendung über die Arbeitsfläche gelegt wird und das nicht verschoben oder rotiert werden kann. Die jeweiligen Informationen werden dabei in einem an der horizontalen Achse gespiegelten Textfeld angezeigt. Die Art des Dialogs (Warnung, Fehlermeldung, Information, Frage) wird durch ein Icon visualisiert. Darüber hinaus wird beim Aufschalten eines Dialogs die Arbeitsfläche für weitere Benutzereingaben gesperrt, um zu gewährleisten, dass der Fokus aller Benutzer auf die durch den Dialog dargestellte Information gerichtet wird.

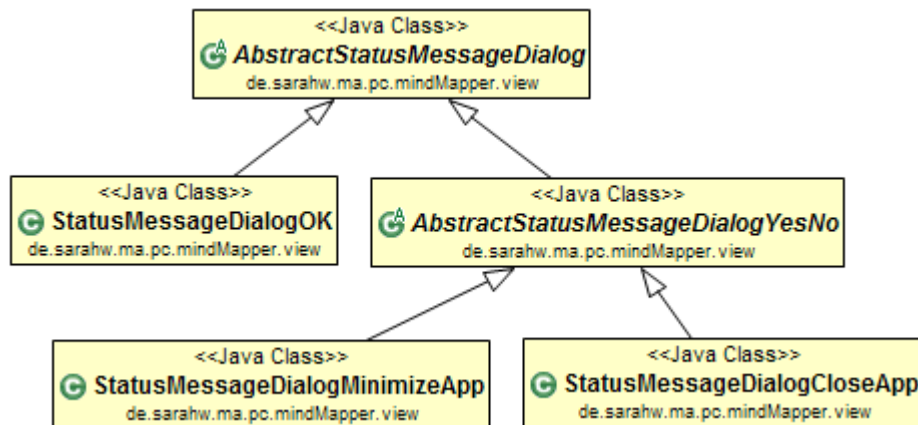


Abbildung 54: Auszug aus dem UML-Klassendiagramm des Pakets `view` um die `StatusMessageDialog`-Klassen.

Dass die Bestätigung eines Dialogs auch durch nur einen Benutzer erfolgen kann, läuft zugegebenermaßen dem in Kapitel 3.2.4 vorgestellten Designziel der kollekt-

tiven Kontrolle über kritische Systemaktionen zuwider, stellt jedoch im Rahmen der im Kontext dieser Arbeit umsetzbaren Möglichkeiten zumindest eine Form dar, die Ausführung ungewollter kritischer Aktionen zu verhindern. Für die Anwendung umgesetzt wurden zwei Formen von Dialogen, zum einen jene, die nur zur Anzeige von Informationen (wie des Erfolgs beim Speichern und Laden oder des Bluetooth-Status nach Start der Anwendung) dienen und durch einen OK-Button geschlossen werden können (Klasse `StatusMessageDialogOK`, abgeleitet von `AbstractStatusMessageDialog`), sowie solche, die die Bestätigung oder den Abbruch einer bestimmten Aktion erzwingen (Subklassen von `AbstractStatusMessageDialogYesNo`) (Abbildung 54 und 55).

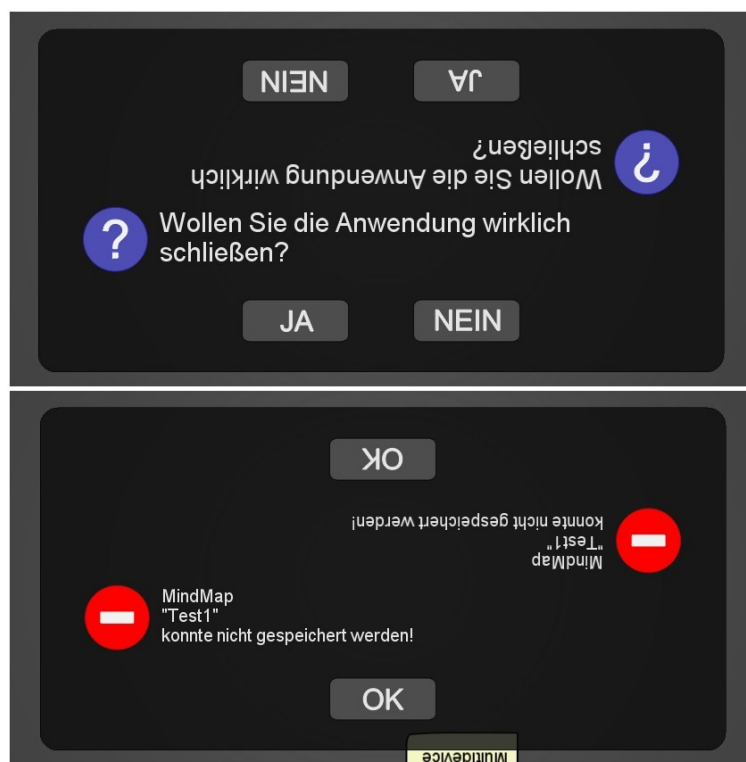


Abbildung 55: Beispieldialoge beim *Multi/Touch/Device MindMapper* (Warnung vor dem Schließen der Anwendung, Fehler beim Laden Speichern).

Weitere Klassen des Pakets `view` umfassen schließlich einige an die Anforderungen der Anwendung angepasste Klassen zur Verarbeitung von Gesten und zur Ausführung von Aktionen durch bestimmte Gesten sowie diverse Hilfsklassen wie Aufzählungstypen, Farbdefinitionen und Sammlungen von allgemein genutzten Methoden. Anschließend wird auch die Paket- und Klassenstruktur der Anwendung für das Smartphone vorgestellt.

Paket- und Klassenstruktur Smartphone

Die Konzeption der Applikation für das Smartphone erfolgte in erster Linie auf Basis der offiziellen Guidelines zur Benutzerschnittstellengestaltung für die gewählte Zielplattform *Android*, wobei primär bereits vom Framework bereitgestellte Widgets verwendet wurden.²⁸⁵ Der geringe Funktionsumfang der Anwendung bedingt letztlich eine Aufteilung in nur zwei Pakete – das Paket *phone*, das alle zentralen Klassen der Anwendung enthält, sowie das Paket *DragNDrop*, das (teilweise modifizierte) Klassen des von Eric Harlow entwickelten *DragNDrop ListView*-Widgets umfasst, welches für den Transfer von Ideentexten mittels *Drag and Drop* eingesetzt wurde.²⁸⁶

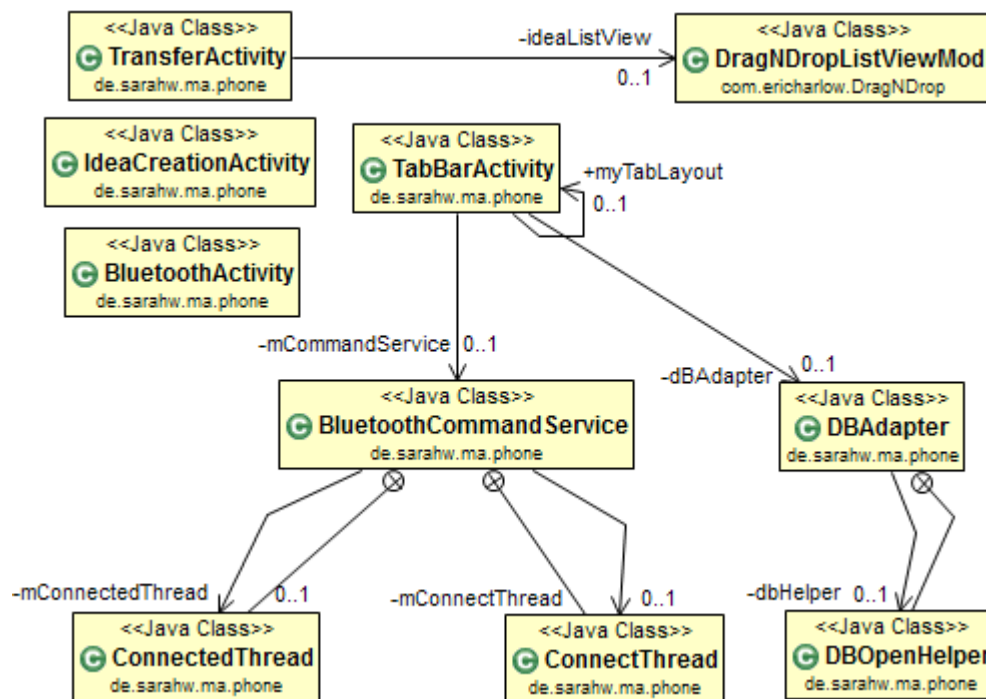


Abbildung 56: Auszug aus dem UML-Klassendiagramm der *Android*-Applikation des *Multi/Touch/Device MindMappers*.

Das Paket *phone* umfasst dabei zum einen die Klassen der drei zentralen *Activities*, welche die im Rahmen des Anwendungskonzept separierten Funktionsbereiche ‚Ideentransfer‘, ‚Ideenverwaltung‘ und ‚Bluetooth‘ repräsentieren. In jeder dieser

²⁸⁵ Vgl. Google Inc. (2011).

²⁸⁶ Vgl. Harlow (2010).

Activities lässt sich zudem über die Menü-Taste des Smartphones ein Hilfedialog aufschalten. Systemmeldungen wie der erfolgreiche Transfer, das Speichern oder das Löschen einer Idee werden über *Toasts*, kleine für eine kurze Zeit sichtbare Overlays, angezeigt. Zum anderen definiert die *Activity* *TabBarActivity* an dieser Stelle das diesen drei Ansichten übergeordnete Tab-Menü. *TabBarActivity* stellt darüber hinaus die Verbindung mit der Datenbank über die Klasse *DBAdapter* und dem im Hintergrund der Anwendung laufenden Bluetooth-Service *BluetoothCommandService* her (Abbildung 56).

Die Benutzerschnittstelle der Android Anwendung des *Multi/Touch/Device MindMappers* besteht aus drei über ein Tab-Widget (*TabBarActivity*, abgeleitet von *TabActivity*) auswählbaren *Activities*: Der *TransferActivity*, der *IdeaCreationActivity* und der *BluetoothActivity* (alle abgeleitet von *Activity*).

Das Erstellen und Löschen von Ideen geschieht über die *IdeaCreationActivity*, welche durch Auswahl des Tabs mit der Bezeichnung ‚Ideenmanager‘ ausgewählt werden kann. Die *Activity* beinhaltet ein Texteingabefeld, einen Button zum Speichern des eingegebenen Texts und ein Listen-Widget, das alle bisher gespeicherten Ideentexte anzeigt. Durch Tippen auf das Texteingabefeld wird die virtuelle Bildschirmtastatur aufgeschaltet (Abbildung 57). Der Benutzer kann daraufhin einen Text mit einer Maximallänge von 30 Zeichen eingeben. Durch die Bestätigung über den Button wird die neue Idee, sofern sie nicht bereits vorhanden ist, in der Datenbank gespeichert und erscheint als neuer Eintrag zu Beginn der Liste. Die Verbindung mit der Datenbank wird dabei durch einen von der *TabBarActivity* instantiierten *DBAdapter* hergestellt (Abbildung 56). Die Verbindung zwischen Datenmodell (Datenbank) und Präsentation (Listen-Widget) erfolgt über einen *SimpleCursorAdapter*.²⁸⁷ Durch ein langes Tippen (*Tap and Hold*) auf einem Listeneintrag kann die entsprechende Idee wieder aus der Datenbank gelöscht werden.

Der Transfer von gespeicherten Ideen kann anschließend über die *TransferActivity* erfolgen, welche durch Auswahl des Tabs ‚Ideentransfer‘ ausgewählt wird. Dieser Screen besteht ebenfalls aus der Liste der gespeicherten Ideen sowie einem dedizierten Bereich, in den einzelne Ideen via *Drag and Drop* gezogen werden können, um den Transfer auf den Tisch anzustoßen.

²⁸⁷ Adaption der Klassen *DBAdapter* und *DBOpenHelper* aus Übungsmaterialien des Kurses *Einführung in die Softwareentwicklung mit Android* im Sommersemester 2011 an der Universität Regensburg.

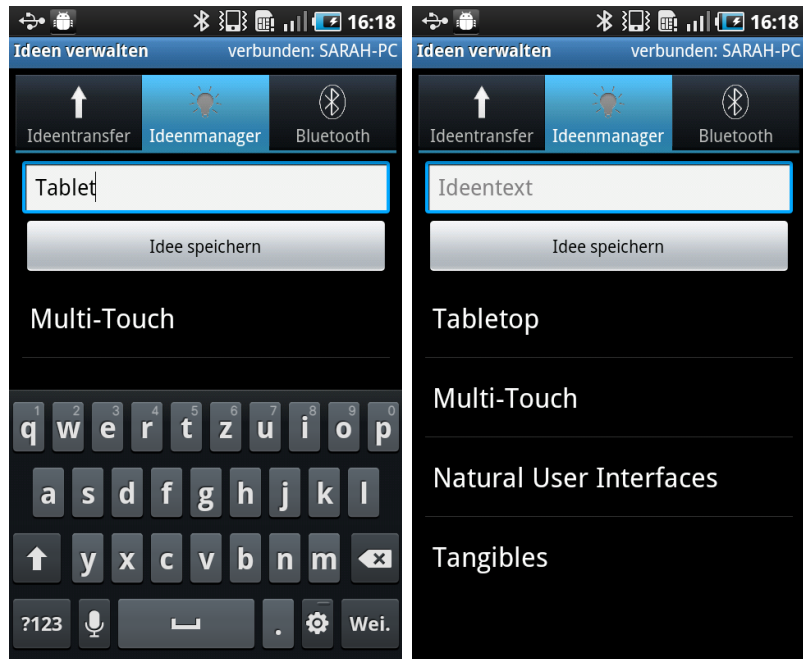


Abbildung 57: Screenshots der IdeaCreationActivity: Eingabe einer neuen Idee (links) und Ansicht ohne virtuelle Tastatur (rechts).

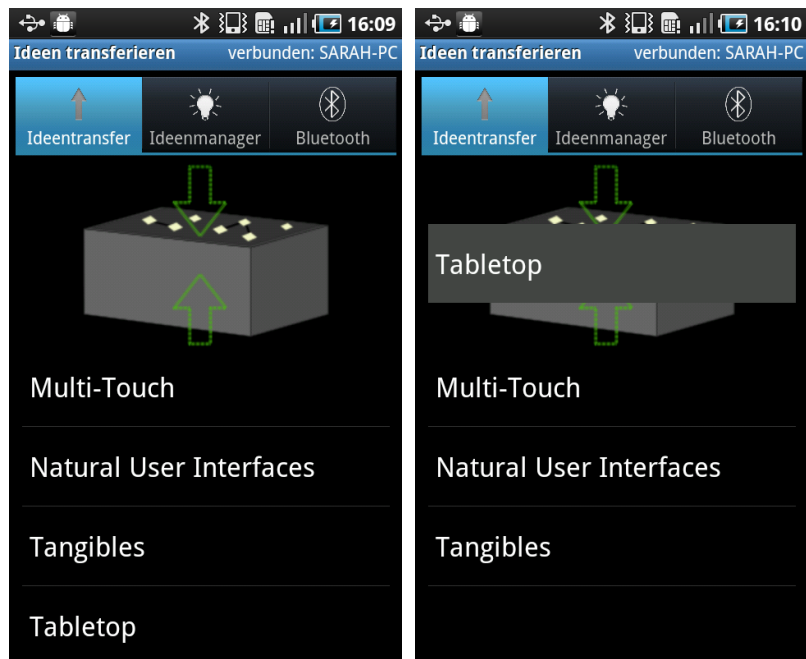


Abbildung 58: Screenshots der TransferActivity: Normale Ansicht (links) und Transfer einer Idee über *Drag and Drop* auf einen bestimmten Bereich (rechts).

Visualisiert wird dieser Bereich durch eine stilisierte Darstellung des Tabletop, welche durch zwei Pfeile signalisiert, wohin der Benutzer eine Idee ziehen kann, um einen Transfer zu erwirken (Abbildung 58).²⁸⁸ Durch die Übertragung eines Ideentextes wird dieser schließlich aus der Datenbank und den entsprechenden Listen entfernt. Da das durch die *Android*-API zur Verfügung gestellte Listen-Widget keine *Drag and Drop*-Funktionalität unterstützt und darüber hinaus *Drag and Drop*-Gesten erst ab *Android 3.0 Honeycomb* Teil des *Android* API sind, wurde hier ein von Eric Harlow modifiziertes Listen-Widget eingesetzt, welches das Verschieben einzelner Einträge innerhalb einer Liste erlaubt.²⁸⁹ Im Rahmen des *Multi/Touch/Device MindMapper* Projekts kommt schließlich mit `DragAndDropListViewMod` (abgeleitet von `ListView`) eine an die Anforderungen der Anwendung angepasste Version dieses Widgets zum Einsatz (Abbildung 56). Während der rechte Bereich der Liste für das Scrollen reserviert ist, ermöglicht das Berühren des restlichen Bereichs eines Listeneintrages dessen Verschieben in vertikale Richtung. Der Transfer erfolgt erst bei Loslassen der Idee über dem dedizierten Transferbereich.

Voraussetzung für eine erfolgreiche Übertragung ist dabei eine bestehende Verbindung mit der Anwendung auf dem Tabletop. Diese muss zunächst manuell über die `BluetoothActivity` hergestellt werden. Die hierfür nötigen Mechanismen sowie einige Elemente der *Activity* wurden analog zur Funktionalität des Bluetooth-Servers am Tabletop aus dem von Luu Gia Thuy entwickelten *Remote Bluetooth Android*-Projekt adaptiert.²⁹⁰ Die `BluetoothActivity` enthält somit neben der Anzeige eines gegebenenfalls verbundenen Gerätes eine Liste der gekoppelten sowie anderer in Reichweite befindlichen Geräte. Letztere wird erst durch Auswahl des Buttons ‚Nach verfügbaren Geräten suchen‘ dynamisch befüllt. An dieser Stelle kommt der von Mark Murphy entwickelte *MergeAdapter* (über eine Bibliothek eingebunden) zum Einsatz, wodurch die Listen-Widgets für gekoppelte und weitere Geräte sowie deren Titel als eine Liste behandelt werden.²⁹¹ Der Einsatz einer der beiden Listen übergeordneten Liste ist hier nötig, da es bei der Darstellung über die üblichen `ListView`-Widgets vorkommen kann, dass die ers-

²⁸⁸ Eine erste Version dieser Grafik umfasst nur *einen* Pfeil, der in Richtung des stilisierten Tabletops zeigt. Diese wurde jedoch im Zuge erster Beobachtungen der Interaktion mit der Smartphone-Anwendung durch die aktuelle Grafik mit zwei Pfeilen ausgetauscht, um die missverständliche Signalwirkung von nur einem Pfeil zu verringern. Siehe dazu ausführlicher Kapitel 7.1.

²⁸⁹ Vgl. Harlow (2010).

²⁹⁰ Vgl. Thuy (2010).

²⁹¹ Vgl. Murphy (2011).

te Liste so viele Einträge hat, dass die Items der zweiten Liste nicht mehr erreichbar sind.

Durch die Auswahl eines Gerätes aus einer dieser beiden Listen kann schließlich eine Verbindung hergestellt werden. Bei einer ersten Verbindung des Smartphones mit dem Tabletop muss dabei zunächst die Bestätigung einer Passphrase zum Koppeln der Geräte erfolgen (Abbildung 59). Das Trennen eines bereits verbundenen Gerätes erfolgt durch Auswahl des Gerätenamens unter ‚Verbundenes Gerät‘ oder durch Beenden der Anwendung. Zusätzlich wird in jeder *Activity* über eine Zustandsleiste oben rechts der aktuelle Verbindungsstatus angezeigt.

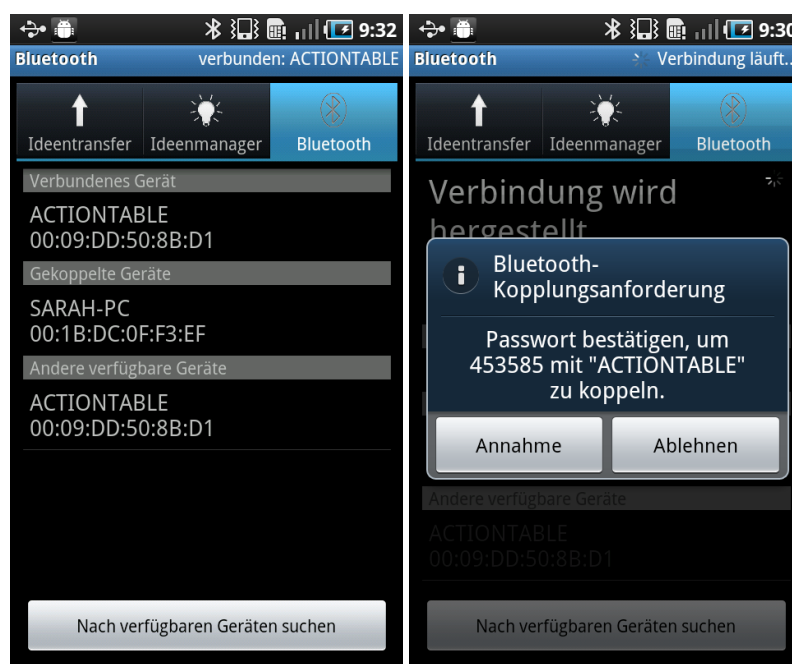


Abbildung 59: Screenshots der BluetoothActivity: Übersicht über Gerätezustände (links) sowie Bluetooth-Kopplungsanforderung (rechts).

Die Kommunikation mit dem lokalen `BluetoothAdapter` des Geräts und das Starten des `BluetoothCommandService`, der für das Herstellen einer Verbindung und den Transfer der Daten über einen `OutputStream` verantwortlich ist, erfolgt dabei ebenfalls über die `TabBarActivity` bei deren Start. Das Auslagern der Bluetooth-Funktionalität in einen Service gewährleistet hier, dass der Benutzer die Anwendung minimieren und parallele Aufgaben, wie die Informationssuche im Internet, durchführen kann, ohne dass die Verbindung zum Tabletop getrennt wird.

Um dem Benutzer die Interaktion mit der Smartphone-Applikation, beispielsweise durch Erklären der *Drag and Drop*-Geste zum Transfer von Ideen oder wie eine Bluetooth-Verbindung herzustellen ist, zu erleichtern, kann zusätzlich für jede *Activity* ein Hilfedialog eingeblendet werden, welcher über das durch die Menü-Taste des Geräts zu aktivierende Kontextmenü zu erreichen ist (Abbildung 60).



Abbildung 60: Hilfedialoge für Ideentransfer (links) und Bluetooth (rechts).

Neben dieser Erläuterung der Paket- und Klassenstruktur sowie der dabei umgesetzten Benutzerschnittstellenelemente der einzelnen Anwendungsteile des *Multi-Touch/Device MindMappers* wird dieses Kapitel abschließend auf diverse Probleme und Konzeptänderungen im Laufe der Implementierungsphase eingehen.

6.3 Konzeptänderungen und Probleme im Laufe der Implementierung

Während der der Konzeptionsphase nachfolgenden Implementierungsphase mussten einige Anpassungen des Konzeptes vorgenommen werden, die sich aus den Möglichkeiten und Grenzen der verwendeten Frameworks sowie zeitlichen Beschränkungen ergaben. In der Folge sei kurz auf während der Implementierung aufgetretene Probleme und einige Konzeptänderungen eingegangen.

Anpassungen des Konzepts

Zunächst erfolgten einige Anpassungen des Konzepts hinsichtlich der Umsetzung des Gestenalphabets. Hier sei vor allem der Wegfall der Funktionen ‚eine Idee selektieren‘ (*Single Tap*) sowie ‚mehrere Ideen selektieren‘ (Lasso-Geste) genannt, welche durch die Konkurrenz mit anderen Gesten im Framework nicht umgesetzt werden konnten. Ein *Single Tap* auf eine Idee wird durch *MT_{4j}* bereits als Teil der *Drag and Drop*-Geste registriert, die Umsetzung der Zustände ‚nicht selektiert‘ und ‚selektiert‘ hätte in diesem Fall einen deutlichen Mehraufwand bedeutet und kann mit Blick auf die eigentlichen Anforderungen an das System vernachlässigt werden. Während die Erkennung einer Lasso-Geste zwar prinzipiell vom *MT_{4j}*-Framework bereitgestellt wird, werden alle Berührungen direkt auf der Arbeitsoberfläche zur Erkennung von Unistroke-Gesten herangezogen, wodurch die Umsetzung der Lasso-Geste zur Selektion von mehreren Ideen nicht mehr möglich ist. Als weitere Einschränkung bei der Umsetzung kann hier auch die Tatsache gesehen werden, dass beim Verbinden zweier Ideen über eine Relation die Pfeilgeste *über* beiden Ideen ausgeführt werden muss. Ein Pfeil *zwischen* zwei Ideen (wie im Anwendungskonzept beschrieben) reicht zur Erstellung einer Relation nicht aus. Des Weiteren verlangt der im *MT_{4j}*-Framework eingesetzte Unistroke-Erkenner das Zeichnen von Symbolen in einem Durchgang ohne Absetzen des Fingers.

Eine weitere Anpassung wurde bezüglich der Größe einzelner Bildelemente vorgenommen. Die erste Umsetzung der durch den maßstabgetreuen Papier-Prototypen visualisierten Widgets zeigte, dass die Interaktion über die Griffe an Menüs und Overlays sich aufgrund deren Größe teilweise schwierig gestaltet. Zudem waren einzelne Texte stellenweise schlecht lesbar. In der Folge wurden somit alle Bildelemente, bis auf die Ideenknoten und Relationen, um ein Drittel vergrößert.

Schließlich musste auch die Option ‚Fenster verkleinern / maximieren‘ gestrichen werden, da diese Funktion im Rahmen einer *MT4j*-Applikation nicht umsetzbar ist.

Probleme während der Implementierungsphase

Neben allgemeinen Anpassungen des Konzepts traten während der Implementierungsphase auch einige Probleme auf, die größtenteils gelöst werden konnten. Bei der Ausführung der Anwendung am *Evoluce ONE* kommt es mit der verwendeten *MT4j*-Version (0.98) häufig zu Abstürzen der *TUIO*-Komponente; was letztlich durch das Einbinden einer alternativen *TUIO*-Bibliothek behoben werden kann.²⁹²

Ein weiteres Problem ist der in *MT4j* implementierte SVG-Renderer, da dieser die Darstellung von Verläufen oder Filtern nicht unterstützt und somit die für die Anwendung umgesetzten Vektorgrafiken nicht völlig korrekt anzeigen kann. Des Weiteren musste auf den Einsatz von Vektorschriften verzichtet werden, da deren Darstellung insbesondere bei kleinen Schriftgrößen sehr stark verpixelt ist. Als Alternative wurde deshalb die Standard-Bitmapschrift von *MT4j* eingesetzt, die zwar bei der Rotation und Verschieben von Textelementen leicht verzerrt, jedoch auch in kleineren Schriftgrößen noch gut lesbar ist.

Eine weitere Einschränkung ist durch die Art und Weise, wie *MT4j* das Zeichnen von grafischen Objekten umsetzt, bedingt. Für den Ladescreen sollte ursprünglich eine Animation eingefügt werden, um dem Benutzer den laufenden Prozess zu visualisieren. Da die Methodenausführung in einer *MT4j*-Anwendung nur über einen einzelnen Thread (*Animation Thread*) erfolgen kann, ist es nicht möglich, gleichzeitig das (Neu-) Zeichnen von Bildelementen im Ladescreen sowie das parallele Instantiieren der *MindMapScene* zu erreichen, weshalb auf eine Animation während der Initialisierung der Anwendung verzichtet werden musste und nur eine statische Grafik zur Visualisierung des Ladevorgangs eingebunden werden konnte.

Ein bisher nicht zufriedenstellend zu lösendes zeigte sich bezüglich der Funktion ‚Fenster minimieren‘ - während dies prinzipiell über die Methode `setExtendedState(Frame.ICONIFIED)` des Java-Frame der *MT4j*-Anwendung erreicht werden kann, hat das Ausführen der Methode am Tabletop (im Gegensatz zu anderen Testrechnern mit gleicher Java-Version und Betriebssystem) keinen Effekt. In der Folge musste ein Workaround implementiert werden, um die für

²⁹² Zu Problemen mit der Version 0.98 von *MT4j* und Abstürzen der *TUIO*-Komponente siehe folgenden Beitrag im *NUIGroup*-Forum: <http://nuigroup.com/forums/viewthread/12836/>

die Kopplung von Smartphones mit dem Tabletop benötigte Möglichkeit des Minimierens des Fensters dennoch umsetzen zu können. Bei Auswahl des Menüpunkts wird aktuell die Tastenkombination *ALT+TAB* ausgeführt, um aus der Anwendung auf den Desktop (oder zu einer anderen offenen Anwendung) zu wechseln, um Zugriff auf das Bluetooth-Icon in der *Windows*-Symbolleiste zu erlangen.

Erweiterungsmöglichkeiten und offene Punkte

Neben allgemeinen Konzeptänderungen sowie Problemen bei der konkreten Umsetzung von Anforderungen mit *MT4j* müssen auch einige Punkte genannt werden, die aus zeitlichen Gründen nicht mehr zufriedenstellend umgesetzt werden konnten.

Zunächst sei hier das eigens entwickelte Texteingabefeld-Widget zu nennen. Hier erfolgt bei einem Zeilenumbruch (durch Drücken von *Enter* oder aufgrund des Erreichens des Zeilenendes) möglicherweise eine Trennung mitten im Wort. Hier wäre eventuell eine Trennung nach vorhandenen Leerzeichen denkbar. Darüber hinaus erfolgt derzeit beim Speichern und Laden keine vorherige Abfrage über einen Dialog, was jedoch wünschenswert wäre, da diese Aktionen ebenfalls kritisch sein können, beispielsweise, weil bereits ein Speicherstand mit dem gleichen Namen existiert oder weil beim Laden eines anderen Speicherstandes die aktuelle Sitzung verloren geht. Ein weiterer offener Punkt ist das Bluetooth-Overlay, das derzeit nur die aktuellen Verbindungen anzeigt, weitere Aktionen sind an dieser Stelle nicht möglich. Auch die Einbindung des QR-Code könnte eventuell sinnvoller an anderer Stelle in der Anwendung geschehen. Die Notwendigkeit, die Anwendung zu minimieren, um ein externes Gerät mit dem Tabletop zu koppeln, stellt schließlich einen weiteren Punkt dar, der gegebenenfalls alternativ innerhalb der Anwendung gelöst werden könnte, um den Arbeitsprozess der anderen Benutzer nicht zu stören.

Letztlich lässt sich festhalten, dass trotz einiger Anpassungen des Konzepts, stellenweisen Problemen in der Umsetzung und diversen offenen Punkten eine erfolgreiche prototypische Umsetzung der Anforderungen geleistet werden konnte. Sowohl die im Rahmen des Konzept definierten Funktionen als auch die in der Folge formulierten Designziele konnten hierbei verwirklicht werden. Da es sich hier um eine prototypische Version der Anwendung handelt, wurde ein besonderes Augenmerk auf Wiederverwendbarkeit und Erweiterbarkeit des Quellcodes gelegt.

Eine zentrale Einschränkung, die sich jedoch aus dem zeitlichen Rahmen der Arbeit ergibt, ist, dass keine umfangreiche Evaluation des Systems durchgeführt werden kann. Dies wäre jedoch sowohl in Hinblick auf die allgemeine Interaktion mit dem System als auch für ein Brainstorming und Mind-Mapping-Szenario in der Gruppe interessant, wobei speziell der Einfluss der zusätzlichen Interaktion über ein externes Gerät mit dem Tabletop zu untersuchen wäre. Dennoch konnten erste Erfahrungen mit dem System aus dem Testbetrieb gesammelt werden, welche im die Arbeit abschließenden Kapitel diskutiert werden.

7 Diskussion

Im folgenden Kapitel wird das Ergebnis der Konzeption und Implementierung diskutiert. An dieser Stelle werden zunächst erste Erfahrungen mit dem System aus dem Testbetrieb in der praktischen Anwendungen zur Sprache kommen. Ein Ausblick auf die mögliche Weiterentwicklung der Anwendung, potentielle Problembereiche sowie Evaluationsmöglichkeiten schließt das Kapitel ab.

7.1 Erste Erfahrungen aus dem Testbetrieb

Obwohl der zeitliche Rahmen der Arbeit keine systematische Evaluation des Systems erlaubt, konnten dennoch erste praktische Erfahrungen mit der Anwendung im Rahmen des im November 2011 am Lehrstuhl für Medieninformatik ausgerichteten *World Usability Day* (WUD) gesammelt werden.²⁹³ Während der Präsentation des *Future Interaction Labs* (FIL) konnten Besucher des WUD neben zahlreichen anderen, am Lehrstuhl entstandenen Projektprototypen auch den *Multi/Touch/Device MindMapper* am *Evoluce ONE*-Tabletop sowie auf zwei Smartphones nach Belieben ausprobieren.

Neben überwiegend positiven Bewertungen des Systems wurden hierbei sowohl bekannte Probleme als auch bisher unbeachtete Aspekte von den einzelnen Benutzern verbalisiert. Bereits während der Entwicklung der Anwendung für den Tabletop wurde klar, dass die Genauigkeit der Erkennung von Berührungspunkten am Tisch bisweilen ungenügend ist. Dies wurde während der Interaktion der Besucher des WUD mit dem System bestätigt. Während die Unistroke-Gesten prinzipiell gut verstanden wurden, gestaltete sich deren Ausführung durch die ungenaue Erkennung schwierig, da oftmals der Berührungspunkt des Benutzers noch während der Ausführung einer Unistroke-Geste verloren wird und die Geste somit nicht komplett ausgeführt werden kann. Festeres Aufdrücken der Finger auf die Oberfläche führt in der Regel zu einer Steigerung der Erkennungsgenauigkeit, jedoch ist dies den meisten Benutzern nicht bewusst, da in der Regel bereits eine leichte Berührung der Tabletopoberfläche zum Auslösen einer Aktion bzw. zur Interaktion ausreicht. Dies wurde insbesondere bei der Möglichkeit des Editieren von Ideen deutlich, da beispielsweise die Positionierung eines Fingers über einer Idee und dessen leichte Bewegung bisweilen als *Double Tap* erkannt wurde (trotz Deaktivierung des *Hover*-Modus). Auch die Erkennungsgenauigkeit und Arbeitsweise des *i\$ Unistroke Recognizer* führten bisweilen zu Verwirrung – insbeson-

²⁹³ Vgl. Lehrstuhl für Medieninformatik Regensburg (2011).

dere die Pfeilgeste erforderte häufig mehrere Versuche, bis sie korrekt erkannt wurde. Zudem wird die Pfeilgeste manchmal als X-Geste erkannt, was zum ungewollten Löschen von Ideenknoten führen kann. Schließlich kam es auch gelegentlich zur ungewollten Erstellung von Ideen auf dem Tabletop, weil manchmal bereits ein *Single Tap* oder eine versehentliche Berührung der Oberfläche vom Unistroke-Erkenner als Rechteck bzw. Kreis erkannt wird.

Bezüglich der Texteingabe am Tabletop über die virtuelle Tastatur gab es keine unmittelbar negativen Rückmeldungen, jedoch wurde zum Beispiel die Frage gestellt, warum man nicht einfach direkt auf der Oberfläche schreiben könne. Auffällig war hingegen, dass dadurch, dass die Tastaturen unabhängig vom zugehörigen Textfeld frei verschiebbar sind, teilweise nicht mehr klar war, welche Bildschirmtastatur zu welcher Idee gehört. Hier wäre gegebenenfalls eine visuelle Assoziation zwischen Tastatur und Textfeld, sei es durch farbliche Kennzeichnung oder eine dünne Verbindungslinie, denkbar.²⁹⁴ Ein Benutzer äußerte sich zudem kritisch bezüglich der gespiegelten Texte innerhalb der Ideenknoten – die doppelte Darstellung wäre eher verwirrend als hilfreich und dadurch, dass die Ideen sowieso frei rotiert werden können, sei auch nur eine Ausrichtung der Texte ausreichend.

Eine weitere interessante Beobachtung konnte bezüglich der Interaktion mit dem Fenstermenü gemacht werden: Benutzer betätigten häufig – sowohl aus Neugier als auch versehentlich – den Button zum Minimieren der Anwendung. Da zu diesem Zeitpunkt keine weitere Bestätigung der Aktion über einen Dialog erfolgen musste, wurde dadurch der Arbeitsprozess der anderen Benutzer erheblich gestört. In der Folge wurde hier ein Dialog zur Bestätigung der Aktion hinzugefügt. Dennoch ist das Minimieren der Anwendung beim Koppeln eines neuen Geräts mit dem Tabletop nötig, was den Arbeitsprozess der Benutzer grundsätzlich negativ beeinflusst. An dieser Stelle ist zudem die Interaktion mit der Maus nötig, da außerhalb der Anwendung durch *Windows 7* keine Touch-Events verarbeitet werden. Hier zeigte sich, wie bereits in der Implementierungsphase, dass die Randeinfassung des *Evoluce ONE*-Tabletop mit 5 cm für das Abstellen und Bedienen von externen Eingabegeräten wie Maus oder Tastatur und das Ablegen von Smartphones definitiv zu schmal ist.

Schließlich wurde die Möglichkeit des Skalierens von Bildelementen, die aus Komplexitäts- und Zeitgründen im *Multi/Touch/Device MindMapper* nicht umgesetzt wurde, von einigen Benutzern vermisst. Dies weist darauf hin, dass einzelne Benutzer, vermutlich aufgrund von Vorerfahrungen, bereits bestimmte

²⁹⁴ Vgl. Cowie (2010, S. 21).

Erwartungen bezüglich der Interaktion mit einem Touchscreen mitbringen, auf die in gewissem Maße reagiert werden sollte.

Neben Problemen und Anregungen, welche die direkte Interaktion mit dem Tabletop betreffen, konnten auch einige interessante Aspekte bei der Interaktion mit den externen Geräten beobachtet werden. Allgemein interagierten nur wenige Benutzer mit den zur Verfügung gestellten Smartphones, was jedoch auch als ein Resultat der allgemeinen Situation verstanden werden darf, da vielen der spontan mit dem Tisch interagierenden Besuchern gar nicht klar war, dass eine zusätzliche Interaktionsmöglichkeit mit dem Tabletop über die auf dem Tisch platzierten Smartphones zur Verfügung stand, da nur durch eine Person zusätzliche Hilfestellungen am Tisch gegeben werden konnten. Des Weiteren zeigten sich stellenweise Probleme bezüglich des Transfers von Ideen auf den interaktiven Tisch. Die stilisierte Grafik des Tabletops in der Activity ‚Ideentransfer‘ wurde dahingehend häufig missverständlich interpretiert, dass Benutzer versuchten, einzelne Ideen über eine in Richtung des *tatsächlichen* Tabletops ausgeführte Wisch-Geste statt via *Drag and Drop* auf den designierten Bereich zu transferieren. Jedoch wurde die Möglichkeit der Hilfedialoge der *Android*-Anwendung, soweit im Rahmen der Beobachtung ersichtlich, nur wenig in Anspruch genommen. Die zu diesem Zeitpunkt über nur einen Pfeil den Transferbereich visualisierende Grafik wurde dennoch in der Folge durch eine überarbeitete Grafik ersetzt, welche durch den Einsatz von zwei Pfeilen diese Ambiguität zu reduzieren versucht (Abbildung 61).

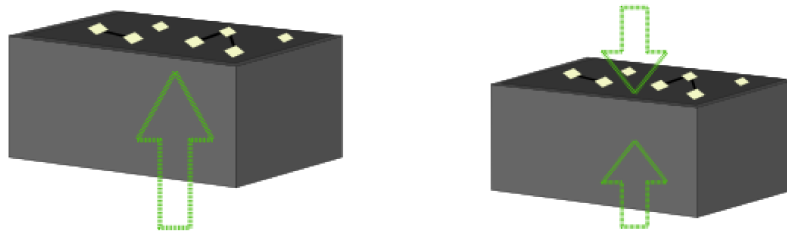


Abbildung 61: Stilisierte Grafik des Tabletops zur Kennzeichnung des Transferbereichs bei der Smartphone-Anwendung: erste Version (links) und überarbeitete Version (rechts)

Darüber hinaus stellten einige Benutzer die Frage, ob man die Ideen auf dem Smartphone auch bearbeiten könne, analog zur Möglichkeit des Editierens auf dem Tabletop. Nicht zuletzt in Hinblick auf die Konsistenz der Interaktionsmöglichkeiten an Smartphone und Tabletop wäre dies eine sinnvolle Ergänzung einer zukünftigen Version des Prototyps. Eine weitere Frage, die häufiger von den Personen, die mit dem Smartphone interagierten, gestellt wurde, bezog sich auf die

Möglichkeit des Transfers einer Idee zurück auf das Smartphone. Derartige bidirektionale Interaktion mit dem Smartphone ist im aktuellen Prototyp bisher nicht umgesetzt, wäre jedoch, wie bereits in Kapitel 5.4 angedeutet, unter der Voraussetzung der Identifizierung einzelner Smartphones durch den Tisch durchaus denkbar.

Die Präsentation des *Multi/Touch/Device MindMappers* im Rahmen des *World Usability Day 2011* konnte somit einige wertvolle Anregungen und Hinweise für die iterative Verbesserung und potentielle Problembereiche der Anwendung liefern. In der Folge sei somit zusammenfassend einer kurzer Ausblick auf Möglichkeiten der Weiterentwicklung, potentielle Problembereiche und Möglichkeiten der Evaluation gegeben.

7.2 Ausblick

Die vorliegende Arbeit zeigt, dass unter anderem mit Hilfe des Frameworks *MT4j* eine effiziente Umsetzung von prototypischen Anwendungen für multitouchfähige Tabletops möglich ist. Darüber hinaus wurde mit Hilfe der *Android* API und weiteren, bereits bestehenden Projekten und Komponenten eine Ergänzung der Anwendung durch Smartphones als zusätzliche Eingabemodalität und zur Schaffung von privaten Räumen ermöglicht. Zentrale Funktionen wie das Erstellen, Bearbeiten und Manipulieren von digitalen Notizzetteln sowie deren Verknüpfung zu Mind-Maps konnten erfolgreich umgesetzt werden. Über diese grundlegenden Anforderungen hinaus ist natürlich neben den in Kapitel 7.1 skizzierten offenen Punkten und unmittelbar umsetzbaren Erweiterungsmöglichkeiten für die bestehende Anwendung weitere Funktionalität denkbar. Kapitel 5.4 beschreibt einige weitere Möglichkeiten, wie der *Multi/Touch/Device MindMapper* in Hinblick auf *Usability* und Funktionsumfang verbessert werden könnte. Diese Ideen lassen sich unter folgende allgemeine Punkte zusammenfassen:

- zusätzliches Feedback in Bezug auf Interaktion, Arbeitsziel und Kollaboration
- Unterstützung wiederverwendbarer Dateiformate (Import/Exportfunktion über XML)
- Alternative Optionen der Verbindung von Smartphone und Tabletop (WLAN)
- Erweiterung von Eingabemodalitäten (Handschriftliche Eingabe, fest verbaute Tastatur am Smartphone)

- Ausschöpfen des Potentials der Multi-Device-Interaktion (bidirektionaler Transfer, erweitertes Feedback)

Neben all diesen Aspekten ist jedoch die zentrale Frage, die sich im Kontext von Brainstorming und Mind-Mapping in einem Multi-Device-Kontext stellt, jene nach der praktischen Relevanz und gegebenenfalls messbaren Vor- und Nachteilen gegenüber anderen Lösungsansätzen im Kontext elektronisch unterstützter Problemlösungsprozesse in Gruppen. Die im Rahmen dieser Arbeit geleistete Konzeption und Umsetzung des *Multi/Touch/Device MindMappers* kann lediglich die technische Grundlage für die Beantwortung dieser Fragen darstellen. Da aufgrund zeitlicher Beschränkungen eine frühe und intensive Einbindung zukünftiger Nutzer des Systems nicht möglich war, wäre eine der Implementierung nachfolgende Evaluation des Systems sowohl in Bezug auf Aspekte wie *Usability* oder *Joy of Use* sowie die Benutzerfreundlichkeit der bisher umgesetzten Designlösungen als auch in Hinblick auf konkrete Fragestellungen bezüglich der Effekte des Multi-Device-Szenarios auf das kollaborative Arbeiten sinnvoll. Dabei sind zunächst grundlegende Annahmen, wie positive Effekte von Multi-Device-Anwendungen in Bezug auf die allgemeine Produktivität der Gruppenteilnehmer – beispielsweise durch die Verringerung negativer sozialpsychologischer Effekte (siehe Kapitel 2.4 und 4.2) – im Anwendungskontext zu überprüfen. Des Weiteren kann das allgemeine Benutzerverhalten – wie viel Zeit verbringen Benutzer am Tisch, wie viel am Smartphone, und welche Aufgaben führen sie dabei durch – von Interesse sein. An dieser Stelle sollte jedoch auch ein besonderes Augenmerk auf den Aspekt der *Group Awareness* gelegt werden, welche eventuell durch den Einsatz von privaten Displays in Form von Smartphones leiden kann, wodurch es wiederum zu reduzierter interpersoneller Kommunikation und Kollaboration kommen kann. Hier wäre beispielsweise ein Evaluationsszenario denkbar, in dem ein Brainstorming bzw. Mind-Mapping-Prozess sowohl mit als auch ohne Smartphones am Tabletop durchgeführt wird. Ebenso von Interesse könnte der Vergleich mit traditionellen, papierbasierten Methoden sein. Ein weiterer interessanter Aspekt ist hier schließlich auch die Effektivität und Benutzerzufriedenheit bei der Texteingabe über virtuelle Tastaturen sowohl am Tabletop als auch am Smartphone.

8 Fazit

Im Rahmen dieser Arbeit konnte neben der Konzeption einer Anwendung zum Brainstorming und Mind-Mapping am Tabletop mit Smartphones zusätzlich auf Basis des Frameworks *MT4j* und der API des Smartphone-Betriebssystems *Android* erfolgreich ein funktionaler *High-Level*-Prototyp umgesetzt werden. Der Beschreibung des Entwurfs und der Implementierung des *Multi/Touch/Device Mind-Mappers* vorangestellt, gibt die Arbeit ebenso einen umfangreichen Überblick über die Entwicklung der Tabletoptechnologie der letzten zwanzig Jahre und die dabei für die Entwicklung von Tabletopbenutzerschnittstellen aktuell relevanten Problem- und Forschungsfelder sowie über Aspekte der Multi-Device-Interaktion.

Die in Kapitel 5 definierten Anforderungen sowie speziell auf den Anwendungskontext bezogene Designziele – darunter beispielsweise die Umsetzung einer einfachen, auf die eigentlichen Inhalte fokussierten Benutzerschnittstelle am Tabletop, die durch geeignetes Feedback zudem die *Group Awareness* zu fördern versucht – konnten, bis auf wenige Ausnahmen, implementiert werden. Neben der Erstellung, dem Editieren, freien Positionieren und Ausrichten von Ideen sowie deren Verknüpfung in Form von Mind-Maps durch Multi-Touch-Gesten am Tabletop ermöglichen mit dem interaktiven Tisch über Bluetooth verbundene Smartphones eine alternative Möglichkeit der textuellen Eingabe von Ideen, welche schließlich mit Hilfe einer *Drag and Drop*-Geste auf den Tabletop transferiert werden können. Ein Smartphone stellt an dieser Stelle zusätzlich einen privaten Arbeitsbereich dar, in dem ein Benutzer neben der Gruppenarbeit am Tabletop (*teamwork*) der für das Erreichen des Arbeitsziels nötigen Individualarbeit (*taskwork*) nachgehen kann, ohne die anderen Gruppenmitglieder in ihrem Arbeitsprozess zu stören.

Motivation für die Umsetzung einer Anwendung zur Generierung und Strukturierung von Ideen in einem Multi-Device-Kontext war dabei unter anderem die Annahme, dass die Ergänzung einer Tabletopanwendung zum Brainstorming und Mind-Mapping um zusätzliche Geräte in Form von Smartphones Potential für das kollaborative Arbeiten in der Gruppe birgt. So wird beispielsweise angenommen, dass der Einsatz von privaten Eingabegeräten die gegebenenfalls in einer Gruppe auftretende Angst vor Bewertung (*evaluation apprehension*) reduzieren helfen kann und zusätzlich einen Ausgleich zwischen der für das Erreichen der Arbeitsaufgabe gleichermaßen wichtigen Gruppenarbeit (*teamwork*) und Individualarbeit (*taskwork*) schaffen kann. Ebenso lässt sich anführen, dass von den unterschiedlichen Möglichkeiten der Individualisierung der Texteingabe am Smart-

phone profitiert werden kann. Eine Überprüfung dieser Annahmen durch Evaluierung der Anwendung im praktischen Einsatz konnte im zeitlichen Rahmen der Arbeit leider nicht geleistet werden, dennoch stellt die Umsetzung des *Multi/Touch/Device MindMapper* einen funktionalen Prototyp dar, der als Basis für die Beantwortung zukünftiger Fragestellungen dieser Art herangezogen werden kann.

Anhang

A – Daten CD

Dieser Arbeit ist eine CD beigelegt, die zusätzliche relevante Daten enthält, darunter unter anderem Quellcode und Distribution der Anwendung *Multi/Touch/Device MindMapper*. Die Ordnerstruktur der CD lässt sich wie folgt beschreiben:

- *Multi/Touch/Device MindMapper:*

In diesem Ordner finden sich der Quellcode, Distributionsdateien der aktuellen Version, die aus dem Quellcode generierte *JavaDoc* und weitere UML-Diagramme der Paket- und Klassenstruktur der Anwendung.

- *Masterarbeit:*

In diesem Ordner finden sich eine digitale Version dieser Arbeit sowie des Vortrags zur Masterarbeit im Oberseminar des Fachs Informationswissenschaft im Sommersemester 2011.

Literatur- und Quellenverzeichnis

Letzte Prüfung der Online-Quellen am 02.01.2012.

Aiken, Milam; Rebman, Carl & Vanjani, Mahesh (2007). Comment Generation with three Electronic Brainwriting techniques. In: *Academy of Information and Management Sciences Journal* 10 (1), S. 11–30.

Altorfer, Karin (2007). *MindMap - Creative and collaborative MindMapping on a Sensitive Whiteboard*. Master Thesis. Universität Freiburg, Schweiz. Institut für Informatik. Online: [<http://diuf.unifr.ch/main/diva/teaching/master-projects/mindmap-creative-and-collaborative-mindmapping-on-a-sensitive-whiteboard>].

Annett, Michelle; Grossman, Tovi; Wigdor, Daniel & Fitzmaurice, George (2011). Medusa: a proximity-aware multi-touch tabletop. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. Santa Barbara, CA, USA — October 16 - 19, 2011. New York: ACM, S. 337–346. Online: [<http://doi.acm.org/10.1145/2047196.2047240>].

Apted, Trent; Collins, Anthony & Kay, Judy (2009). Heuristics to support design of new software for interaction at tabletops. In: *CHI 2009 Workshop - Multi-touch and Surface Computing*. Online: [http://sydney.edu.au/engineering/it/~judy/Homec/Pubs/2009_CHI_wkshop_heuristics].

Baraldi, Stefano; Bimbo, Alberto & Valli, Alessandro (2006). Bringing the Wiki Collaboration Model to the Tabletop World. In: *2006 IEEE International Conference on Multimedia and Expo (ICME '06)*, S. 333–336. Online: [<http://doi.ieeecomputersociety.org/10.1109/ICME.2006.262466>].

Bau, Olivier; Poupyrev, Ivan; Israr, Ali & Harrison, Chris (2010). TeslaTouch: electrovibration for touch surfaces. In: *Proceedings of the 23rd annual ACM symposium on User interface software and technology (UIST '10)*. New York, NY, USA — October 03 - 06, 2010. New York: ACM, S. 283–292. Online: [<http://doi.acm.org/10.1145/1866029.1866074>]

Benko, Hrvoje & Wigdor, Daniel (2010). Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays. (Human-Computer Interaction Series)*. London: Springer, S. 249–275.

- Boldrin, Massimo (2011).** *Reactable Live! Press Ready Pictures*. Reactable Systems. Online :
[http://www.reactable.com/files/reactable_live_pictures_massimo_boldrin.zip].
- Buisine, Stéphanie; Besacier, Guillaume; Najm, Marianne; Aoussat, Améziane; Vernier, Frédéric (2007).** Computer-supported creativity: Evaluation of a tabletop mind-map application. In: *Proceedings of the 7th international conference on Engineering psychology and cognitive ergonomics (EPCE '07)*. Heidelberg: Springer, S. 22–31. Online: [<http://www.limsi.fr/Individu/besacier/papers/07-buisine-epce.pdf>].
- Buxton, William (1990).** A three-state model of graphical input. In: *Human-Computer Interaction - INTERACT '90 Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*. Cambridge, UK, 27–31 August, 1990. Amsterdam: Elsevier Science Publishers B.V., S. 449–456. Online: [<http://www.billbuxton.com/3state.html>].
- Buzan, Tony & Buzan, Barry (2002).** *Das Mind-map-Buch. Die beste Methode zur Steigerung Ihres geistigen Potenzials*. München: MVG Verlag.
- Cao, Xiang; Wilson, Andrew D.; Balakrishnan, Ravin; Hinckley, Ken & Hudson, Scott E. (2008).** ShapeTouch: Leveraging contact shape on interactive surfaces. In: *Third IEEE International Workshop on Tabletops and Interactive Surfaces (Tabletop 2008)*. Amsterdam, The Netherlands October 1–3, 2008. S. 129–136. Online : [http://www.cs.toronto.edu/~caox/tabletop2008_shapetouch.pdf].
- Chehimi, Fadi & Rukzio, Enrico (2010).** Throwing Gesture as a Way for Photo Sharing between Mobile Phones and Interactive Tables. In: *Workshop at the 4th Intermedia Open Forum*. Palma de Mallorca, September 1, 2010. Online: [http://www.comp.lancs.ac.uk/~rukzio/publications/intermedia2010_chehimi.pdf].
- Cheng, Kelvin; Müller-Tomfelde, Christian & Ruiz, Natalie (2010).** Interactive tabletops with non-interactive rims. In: *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction (OZCHI '10)*. Brisbane, Australia — November 22 - 26, 2010. New York: ACM, S. 296–299. Online: [<http://doi.acm.org/10.1145/1952222.1952285>].
- Circle Twelve Inc (2008).** *DiamondTouch Product Flyer*. Online: [<http://www.circletwelve.com/circle12/images/file/DTflier-web.pdf>].

- Clayphan, Andrew; Collins, Anthony; Ackad, Christopher James; Kummerfeld, Bob & Kay, Judy (2011).** *Firestorm: a brainstorming application for collaborative group work at tabletops*. Technical Report Nr. 678- School of Information Technologies, University of Sydney. Online: [<http://sydney.edu.au/engineering/it/research/tr/tr678.pdf>].
- Collaros, Panayiota A.; Anderson, Lynn R. (1969).** Effect of perceived expertness upon creativity of members of brainstorming groups. In: *Journal of Applied Psychology* 53 (2, Part 1), S. 159–163.
- Cowie, Aaron (2010).** *Collaborative Concept Mapping at an Interactive Tabletop*. Bachelor Thesis. School of Information Technologies, University of Sydney. Online: [http://sydney.edu.au/engineering/it/~judy/Homeec/Theses/2010_Cowie.pdf].
- Cravotta, Robert (2011).** *Touch with the Microsoft Surface 2.0*. Embedded Insights Inc. Online: [<http://www.embeddedinsights.com/channels/2011/03/29/touch-with-the-microsoft-surface-2-0/>].
- Da Luz, T.; Loup-Escande, E.; Christofol, H.; Richir, S. (2010).** The Collaborative Product Design and Help to Decision Making: Interactive Mind-Mapping. In: *Global Product Development. Proceedings of the 20th CIRP Design Conference*, Ecole Centrale de Nantes, Nantes, France, 19th-21st April 2010, S. 237–244.
- Davies, Martin (2011).** Concept mapping, mind mapping and argument mapping: what are the differences and do they matter? In: *Higher Education* 62 (3), S. 279–301. Online: [<http://dx.doi.org/10.1007/s10734-010-9387-6>].
- Deininghaus, Stephan; Möllers, Max; Wittenhagen, Moritz & Borchers, Jan (2010).** Hybrid Documents Ease Text Corpus Analysis For Literary Scholars. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. Saarbrücken, Germany — November 07 - 10, 2010. New York: ACM. Online: [<http://www.dfki.de/its2010/papers/fp246.html>].
- Dennis, Alan R.; George, Joey F.; Jessup, Len M.; Nunamaker, J. F.; Vogel, Douglas (1988).** Information Technology to Support Electronic Meetings. In: *Management Information Systems Quarterly* 12 (4), S. 591–624.
- Diehl, Michael; Stroebe, Wolfgang (1987).** Productivity Loss In Brainstorming Groups: Toward the Solution of a Riddle. In: *Journal of Personality & Social Psychology* 53 (3), S. 497–509.

- Dietz, Paul & Leigh, Darren (2001).** DiamondTouch: a multi-user touch technology. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST '01)*. Orlando, FL, USA — November 11 - 14, 2001. New York: ACM, S. 219–226. Online: [<http://doi.acm.org/10.1145/502348.502389>].
- Donker, Hilko & Meixner, André (2009).** Unterstützung von Kreativitätssitzungen an einem Tablet-System. In: Hartmut Wandke (Hrsg.). *Grenzenlos frei!? Proceedings Mensch und Computer 2009. Neunte fachübergreifende Konferenz an der Humboldt-Universität zu Berlin*. München: Oldenbourg, S. 83–92. Online: [http://mc.informatik.uni-hamburg.de/konferenzbaende/mc2009/1_beitraege/mc2009_09_donker.pdf]
- Dourish, Paul & Bellotti, Victoria (1992).** Awareness and coordination in shared workspaces. In: *Proceedings of the 1992 ACM conference on Computer supported cooperative work (CSCW '92)*. Toronto, ON, Canada — November 01 - 04, 1992. New York: ACM, S. 107–114. Online: [<http://doi.acm.org/10.1145/143457.143468>].
- Dugosh, Karen Leggett; Paulus, Paul B.; Roland, Evelyn J.; Yang, Huei-Chuan (2000).** Cognitive Stimulation in Brainstorming. In: *Journal of Personality & Social Psychology* 79 (5), S. 722–735.
- Dunlop, Mark David & Masters, Michelle Montgomery (2009).** Pickup Usability Dominates: A Brief History of Mobile Text Entry Research and Adoption. In: *International Journal of Mobile Human Computer Interaction* 1 (1), S. 42–59. Online: [http://www.cis.strath.ac.uk/cis/research/publications/papers/strath_cis_publication_2268.pdf].
- Eddy, Nathan (2011).** *Apple, Android Comprise 82% of Smartphone Market Share: NPD Group*. eWeek. Online: [<http://www.eweek.com/c/a/Mobile-and-Wireless/Apple-Android-Compose-82-of-Smartphone-Market-Share-NPD-Group-504538/>].
- Eilebrecht, Karl & Starke, Gernot (2010).** *Patterns kompakt. Entwurfsmuster für effektive Software-Entwicklung*. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag.
- Ellis, Clarence A.; Gibbs, Simon J. & Rein, Gail (1991).** Groupware: some issues and experiences. In: *Communications of the ACM* 34 (1), S. 39–58. Online: [<http://doi.acm.org/10.1145/99977.99987>].
- Encyclopedia of Management (2006).** Brainstorming. In: Helms, Marilyn M. & Cengage, Gale (Hrsg.). *Encyclopedia of Management*. Online: [<http://www.enotes.com/management-encyclopedia/brainstorming>].

- Everitt, Katherine (2003).** *UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces*. Mitsubishi Electronic Research Laboratories (MERL), Cambridge, MA. Technical Report 2003-49. Online: [<http://www.merl.com/papers/docs/TR2003-49.pdf>].
- Evoluce (2010).** *Evoluce ONE*. Online: [<http://web.archive.org/web/20100813022524/http://www.evoluce.com/downloads/docs/Multitouch-LCD-Evoluce-ONE.pdf>].
- Evoluce (2011).** *Evoluce TWO – Multitouch via 3D-sensing and Swipe Functionality*. Online: [<http://www.evoluceblog.com/2011/03/evoluce-two-multitouch-via-3d-sensing-and-swipe-functionality/>].
- Findlater, Leah; Wobbrock, Jacob O. & Wigdor, Daniel (2011).** Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In: *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11)*. Vancouver, BC, Canada — May 07 - 12, 2011. New York: ACM, S. 2453–2462. Online: [<http://doi.acm.org/10.1145/1978942.1979301>].
- Finke, Matthias; Kaviani, Nima; Wang, Ivy; Tsao, Vincent; Fels, Sidney & Lea, Rodger (2010).** Investigating Distributed User Interfaces across Interactive Large Displays and Mobile Devices. In: *Proceedings of the Workshop on coupled display visual interfaces (PPD '10)*. In conjunction with AVI '10, S. 16–20. Online: [<http://www.hitlab.utas.edu.au/w/images/e/ea/FinalProceedingsPPD10.pdf>].
- Fitzmaurice, George W.; Ishii, Hiroshi & Buxton, William A. S. (1995).** Bricks: laying the foundations for graspable user interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '95)*. Denver, CO, USA — May 07 - 11, 1995. New York: ACM/Addison-Wesley, S. 442–449. Online: [<http://dx.doi.org/10.1145/223904.223964>].
- Fjermestad, Jerry & Hiltz, Starr Roxanne (1998).** An assessment of group support systems experimental research: methodology and results. In: *Journal of Management Information Systems - Special issue: GSS insights: a look back at the lab, a look forward from the field* 15 (3), S. 7–149. Online: [<http://web.njit.edu/~jerry/Research/Fjermestad-Hiltz-1999.pdf>].
- Forlines, Clifton; Shen, Chia & Buxton, Bill (2005).** Glimpse: a novel input model for multi-level devices. In: *CHI '05 extended abstracts on Human factors in computing systems (CHI EA '05)*. Portland, OR, USA — April 02 - 07, 2005. New York: ACM, S. 1375–1378. Online: [<http://doi.acm.org/10.1145/1056808.1056920>].

- Forlines, Clifton; Wigdor, Daniel; Shen, Chia & Balakrishnan, Ravin (2007).** Direct-touch vs. mouse input for tabletop displays. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)*. San Jose, CA, USA — April 30 - May 03, 2007. New York: ACM, S. 647–656. Online: [<http://doi.acm.org/10.1145/1240624.1240726>].
- Forster, Florian J. M. (2010).** *Computerunterstützung von kollaborativen Kreativitätsprozessen*. Dissertation. Technische Universität München, Fakultät für Informatik. Online: [<http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100301-818195-1-9>].
- Frisch, Mathias; Heydekorn, Jens & Dachsel, Raimund (2009).** Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. Calgary, AB, Canada — November 23 - 25, 2009. New York: ACM, S. 149–156. Online: [<http://doi.acm.org/10.1145/1731903.1731933>].
- Fukumoto, Masaaki & Sugimura, Toshiaki (2001).** Active Click: Tactile Feedback for Touch Panels. In: *CHI '01 extended abstracts on Human factors in computing systems (EA CHI '01)*. Seattle, WA, USA — March 31 - April 05, 2001. New York: ACM, S. 121–122. Online: [<http://doi.acm.org/10.1145/634067.634141>].
- Gallupe, R. Brent; Bastianutti, Lana M. & Cooper, William H. (1991).** Unblocking Brainstorms. In: *Journal of Applied Psychology* 76 (1), S. 137–142. Online: [<http://dx.doi.org/10.1037/0021-9010.76.1.137>].
- Gerken, Jens; Jetter, Hans-Christian; Schmidt, Toni & Reiterer, Harald (2010).** Can “Touch” Get Annoying? In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. Saarbrücken, Germany — November 07 - 10, 2010. New York: ACM. Online: [<http://www.dfki.de/its2010/papers/poi86.html>].
- Geschka, Horst (2007).** Kreativitätstechniken. In: Richard Köhler (Hrsg.): *Handwörterbuch der Betriebswirtschaft*. Stuttgart: Schäffer-Poeschel, S. 992–1005.
- Geyer, Florian; Pfeil, Ulrike; Budzinski, Jochen; Höchtl, Anita & Reiterer, Harald (2011).** Affinitytable - a hybrid surface for supporting affinity diagramming. In: *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III (INTERACT'11)*. Heidelberg: Springer, S. 477–484. Online: [http://hci.uni-konstanz.de/downloads/FG_AffinityTable_Springer.pdf].

- Google Inc. (2011).** *Android Developers: User Interface Guidelines*. Online: [http://developer.android.com/guide/practices/ui_guidelines/index.html].
- Graham, Nicholas; Scott, Stacey; Gutwin, Carl; Maurer, Frank; Damm, Steve; Phan, Andy & Seyed, Teddy (2010).** *TableTop Widgets: A customizable widget library designed for Tabletop Applications*. SurfNet Wiki. University of Calgary. Online: [<http://www.nsercsurfnet.ca/pmwiki.php?n=SurfNet.GrahamGutwin-MaurerTTWidgets>].
- Greenberg, Saul; Boyle, Michael; Laberge, Jason (1999).** PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. In: *Personal and Ubiquitous Computing 3* (1-2), S. 54-64. Online: [<http://dx.doi.org/10.1007/BF01305320>]
- Gross, Tom & Koch, Michael (2007).** *Computer-supported cooperative work*. München: Oldenbourg.
- Gryskiewicz, Stanley S. (1984).** Uniformity pressure revisited: An evaluation of three creative problem solving techniques in an industrial setting. In: *Division 14 Symposium, Creativity in the Corporation; Ninety-Second Annual Convention of the APA*, Toronto, Canada.
- Gutwin, Carl & Greenberg, Saul (1998a).** Design for individuals, design for groups: tradeoffs between power and workspace awareness. In: *Proceedings of the 1998 ACM conference on Computer supported cooperative work (CSCW '98)*. Seattle, WA, USA — November 14 - 18, 1998. New York: ACM, S. 207-216. Online: [<http://doi.acm.org/10.1145/289444.289495>].
- Gutwin, Carl & Greenberg, Saul (1998b).** Effects of awareness support on groupware usability. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '98)*. Los Angeles, CA, USA — April 18 - 23, 1998. New York: ACM/Addison-Wesley Publishing Co., S. 511-518. Online: [<http://dx.doi.org/10.1145/274644.274713>].
- Habelski, Stefan (2004).** *Realisation of Territory-Based Interaction Techniques for Supporting Tabletop Collaboration*. Internship Thesis. Universität Calgary / Otto-von-Guericke-Universität Magdeburg. Institut für Simulation und Graphik. Online: [http://innovis.cpsc.ucalgary.ca/Publications/_145].

- Haller, Michael; Leitner, Jakob; Seifried, Thomas; Wallace, James R.; Scott, Stacey D.; Richter, Christoph; Brandl, Peter; Gokcezade, Adam & Hunter, Seth (2010).** The NiCE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings. In: *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. Atlanta, GA, USA — April 10 - 15, 2010. New York: ACM, S. 609–618. Online: [<http://doi.acm.org/10.1145/1753326.1753418>].
- Han, Jefferson Y. (2005).** Low-cost multi-touch sensing through frustrated total internal reflection. In: *Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST '05)*. Seattle, WA, USA — October 23 - 27, 2005. New York: ACM, S. 115–118. Online: [<http://doi.acm.org/10.1145/1095034.1095054>].
- Harlow, Eric (2010).** *Experience - Android Drag and Drop List*. Online: [<http://ericharlow.blogspot.com/2010/10/experience-android-drag-and-drop-list.html>].
- Hesselmann, Tobias; Flöring, Stefan & Schmitt, Marwin (2009).** Stacked Half-Pie menus: navigating nested menus on interactive tabletops. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. Calgary, AB, Canada — November 23 - 25, 2009. New York: ACM, S. 17–24. Online: [<http://doi.acm.org/10.1145/1731903.1731936>].
- Hilliges, Otmar; Terrenghi, Lucia; Boring, Sebastian; Kim, David; Richter, Hendrik & Butz, Andreas (2007).** Designing for collaborative creative problem solving. In: *Proceedings of the 6th conference on Creativity & cognition (C&C'07)*. Washington, DC, USA — June 13 - 15, 2007. New York: ACM, S. 137–146. Online: [<http://doi.acm.org/10.1145/1254960.1254980>].
- Hilliges, Otmar; Baur, Dominikus & Butz, Andreas (2007).** Photohelix: Browsing, Sorting and Sharing Digital Photo Collections. In: *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*. Newport, Rhode Island October 10–October 12. Newport, Rhode Island, USA, S. 87–94. Online: [<http://doi.ieeecomputersociety.org/10.1109/TABLETOP.2007.14>]
- Hinrichs, Uta; Hancock, Mark; Carpendale, Sheelagh & Collins, Christopher (2007).** Examination of Text-Entry Methods for Tabletop Displays. In: *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*. Newport, Rhode Island October 10–October 12. S. 105–112. Online: [<http://doi.ieeecomputersociety.org/10.1109/TABLETOP.2007.10>].

- Hinrichs, Uta & Carpendale, Sheelagh (2011).** Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits. In: *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11)*. Vancouver, BC, Canada — May 07 - 12, 2011. New York: ACM, S. 3023–3032. Online: [<http://doi.acm.org/10.1145/1978942.1979391>].
- Hofer, Ramon; Naeff, Daniel & Kunz, Andreas (2009).** FLATIR: FTIR multi-touch detection on a discrete distributed sensor array. In: *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI '09)*. Regent, United Kingdom — February 16 - 18, 2009. New York: ACM, S. 317–322. Online: [<http://doi.acm.org/10.1145/1517664.1517730>].
- Holz, Christian & Baudisch, Patrick (2011).** Understanding touch. In: *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11)*. Vancouver, BC, Canada — May 07 - 12, 2011. New York: ACM, S. 2501–2510. Online: [<http://doi.acm.org/10.1145/1978942.1979308>].
- Hunter, Seth & Maes, Pattie (2008).** *WordPlay: A Table-Top Interface for Collaborative Brainstorming and Decision Making*. Unveröffentlichtes Paper. Online: [http://ambient.media.mit.edu/people/seth/publications/WordPlayFinalIEEE_AffiliationIncluded.pdf].
- Hutchings, Heather M. & Pierce, Jeffrey S. (2006).** Understanding the wheathers, hows, and whys of divisible interfaces. In: *Proceedings of the working conference on Advanced visual interfaces (AVI '06)*. Venice, Italy — May 23 - 26, 2006. New York: ACM, S. 274–277. Online: [<http://doi.acm.org/10.1145/1133265.1133320>].
- Idziorek, Michael (2007).** *Tangible Tabletop Gaming. Die Ebene zwischen Brett- und Videospiel*. Masterarbeit. Wien: Technische Universität Wien, Institut für Gestaltungs- und Wirkungsforschung. Online: [<http://www.media.tuwien.ac.at/v.maquil/publications/idziorekmasterthesis.pdf>].
- Illik, Johann Anton (2007).** *Verteilte Systeme. Architekturen und Software-Technologien*. Renningen: expert-Verlag.
- Inami, Masahiko; Sugimoto, Maki; Thomas, Bruce H. & Richter, Jan (2010).** Active Tangible Interaction. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. London: Springer, S. 171–187.

- Inkpen, Kori; Hawkey, Kirstie; Kellar, Melanie; Mandryk, Regan; Parker, Karen; Reilly, Derek; Scott, Stacey & Whalen, Tara (2005).** Exploring Display Factors that Influence Co-Located Collaboration: Angle, Size, Number, and User Arrangement. In: *Proceedings of HCI International 2005*. Las Vegas, Nevada, USA, 22-27 July 2005. Online: [http://www.cs.dal.ca/~hawkey/HCI2005_Inkpen.pdf].
- Isaksen, Scott G. (1998).** A Review of Brainstorming Research: Six Critical Issues For Inquiry. In: *Monograph* (302). Online: [<http://www.cpsb.com/resources/downloads/public/302-Brainstorm.pdf>].
- Isaksen, Scott G.; Gaulin, John P. (2005).** A Reexamination of Brainstorming Research: Implications for Research and Practice. In: *Gifted Child Quarterly* 49 (4), S. 315–329. Online: [<http://www.cpsb.com/research/articles/creative-problem-solving/Reexamination-of-Brainstorming-Research.pdf>].
- Ishii, Hiroshi & Ullmer, Brygg (1997).** Tangible bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '97). Atlanta, GA, USA — March 22 - 27, 1997. New York: ACM. Online: [<http://doi.acm.org/10.1145/258549.258715>].
- Jansen, Yvonne; Karrer, Thorsten & Borchers, Jan (2010).** MudPad: Tactile Feedback and Haptic Texture Overlay for Touch Surfaces. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '10). Saarbrücken, Germany — November 07 - 10, 2010. New York: ACM. Online: [<http://www.dfki.de/its2010/papers/sp171.html>].
- Jetter, Hans-Christian; Gerken, Jens & Reiterer, Harald (2010).** Natural User Interfaces: Why We Need Better Model-Worlds, Not Better Gestures. In: *CHI 2010 Workshop - Natural User Interfaces: The Prospect and Challenge of Touch and Gestural Computing*. Atlanta, USA, April 2010. Online: [http://hci.uni-konstanz.de/downloads/NUI2010_CJ_JG_HR.pdf].
- Jordà, Sergi; Geiger, Günter; Alonso, Marcos & Kaltenbrunner, Martin (2007).** The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In: *Proceedings of the 1st international conference on Tangible and embedded interaction* (TEI '07). Baton Rouge, LA, USA — February 15 - 17, 2007. New York: ACM, S. 139–146. Online: [<http://doi.acm.org/10.1145/1226969.1226998>].

- Kammer, Dietrich; Keck, Mandy; Freitag, Georg & Wacker, Markus (2010).** Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features. In: *ACM SIGCHI Symposium on Engineering Interactive Computing Systems: Workshop on Engineering Patterns for Multi-Touch Interfaces*. June 20, 2010, Berlin. Online: [http://vi-c.de/vic/sites/default/files/Taxonomy_and_Overview_of_Multi-touch_Frameworks_Revised.pdf]
- Kammer, Dietrich; Lamack, Frank & Groh, Rainer (2010).** Enhancing the Expressiveness of Fingers: Multi-touch Ring Menus for Everyday Applications. In: *Proceedings of the First International Joint Conference on Ambient Intelligence (Lecture Notes In Computer Science, 6439/2010)*. November 10th - 12th 2010, Málaga, Spain, S. 259-264. Online: [http://dx.doi.org/10.1007/978-3-642-16917-5_27].
- Karau, Steven J.; & Williams, Kipling D. (1993).** Social Loafing: A Meta-Analytic Review and Theoretical Integration. In: *Journal of Personality & Social Psychology* 65 (4), S. 681-706.
- Kerr, Norbert L. (1983).** Motivation losses in small groups: A social dilemma analysis. In: *Journal of Personality & Social Psychology* 45 (4), S. 819-828.
- Kim, David (2006).** *BrainStorm: Entwurf und Implementierung einer direct-touch Schnittstelle für kreative Problemlösungen in lokalen Arbeitsgruppen*. Projektarbeit. München: Ludwig-Maximilians Universität München. Online : [<http://wiki.medien.ifi.lmu.de/view/Main/DavidKim>].
- Kim, David; Dunphy, Paul; Briggs, Pam; Hook, Jonathan; Nicholson, John W.; Nicholson, James & Olivier, Patrick (2010).** Multi-touch authentication on tabletops. In: *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. Atlanta, GA, USA — April 10 - 15, 2010. New York: ACM, S. 1093-1102. Online: [<http://doi.acm.org/10.1145/1753326.1753489>].
- Koike, Hideki; Sato, Toshiki; Nishikawa, Wataru & Fukuchi, Kentaro (2010).** Hand and Object Recognition in Liquid Crystal Displays. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. London: Springer, S. 131-146.
- Kray, Christian; Nesbitt, Daniel; Dawson, John; Rohs, Michael (2010).** User-defined gestures for connecting mobile phones, public displays, and tabletops. In: *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services (MobileHCI'10)*. Lisbon, Portugal — September 07 - 10, 2010. New York: ACM, S. 239-248. Online: [<http://doi.acm.org/10.1145/1851600.1851640>].

Kruger, Russell & Carpendale, Sheelagh (2002). Orientation and Gesture on Horizontal Displays. In: *UbiComp 2002 Workshop on Collaboration with Interactive Walls and Tables*. Online: [<http://www.ipsi.fraunhofer.de/ambiente/collabtablewallws/papers/Carpendale,%20Calgary,%20kruger-carpendale.pdf>].

Kruger, Russell; Carpendale, Sheelagh; Scott, Stacey D. & Greenberg, Saul (2003). How people use orientation on tables: comprehension, coordination and communication. In: *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work* (GROUP '03). Sanibel Island, FL, USA — November 09 - 12, 2003. New York: ACM, S. 369-378. Online: [<http://doi.acm.org/10.1145/958160.958219>].

Lamm, Helmut & Trommsdorff, Gisela (1973). Group versus individual performance on tasks requiring ideational proficiency (brainstorming): A review. In: *European Journal of Social Psychology* 3 (4), S. 361-388. Online: [<http://dx.doi.org/10.1002/ejsp.2420030402>].

Laufs, Uwe; Ruff, Christopher & Weisbecker, Anette (2010). MT4j - An open source platform for multi-touch software development. In: *VIMation Journal* (1), S. 58-64. Online: [<http://vimation.eu/files/Laufs.pdf>].

Laufs, Uwe; Ruff, Christopher & Zibuschka, Jan (2010). MT4j - A Cross-platform Multi-touch Development Framework. In: *ACM SIGCHI Symposium on Engineering Interactive Computing Systems: Workshop on Engineering Patterns for Multi-Touch Interfaces*. June 20, 2010, Berlin. Online: [<http://wiki.iao.fraunhofer.de/images/studien/mt4j-a-cross-platform-multi-touch-development-framework.pdf>].

Lehrstuhl für Medieninformatik Regensburg (2011). *World Usability Day 2011*. Online: [<http://www.uni-regensburg.de/sprache-literatur-kultur/medieninformatik/wud/index.html>]

Leithinger, Daniel; Haller, Michael (2007). Improving Menu Interaction for Cluttered Tabletop Setups with User-Drawn Path Menus. In: *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems* (TABLETOP'07). Newport, Rhode Island October 10-October 12. S. 121-128. Online: [<http://doi.ieeecomputersociety.org/10.1109/TABLETOP.2007.24>].

- Lepinski, G. Julian; Grossman, Tovi & Fitzmaurice, George (2010a).** The design and evaluation of multitouch marking menus. In: *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. Atlanta, GA, USA — April 10 - 15, 2010. New York: ACM, S. 2233-2242. Online: [<http://doi.acm.org/10.1145/1753326.1753663>].
- Lepinski, G. Julian; Grossman, Tovi & Fitzmaurice, George (2010b).** *The design and evaluation of multitouch marking menus – Presentation Video*. Autodesk Research. Online: [http://www.autodeskresearch.com/videos/chi2010_mtmm.mov].
- Le Hong, Sylvia & Biesterfeldt, Jakob (2010).** *Weltweit berührt – Studie zur Untersuchung kultureller Unterschiede und Gemeinsamkeiten bei der gestenbasierten Bedienung von Multitouch-Oberflächen*. Online: [<http://www.uid.com/de/aktuelles/publikationen.html>].
- Mandl, Heinz & Fischer, Frank (2000).** Mapping-Techniken in Lern- und Kooperationsprozessen. In: Mandl, Heinz & Fischer, Frank (Hrsg.). *Wissen sichtbar machen. Mapping-Techniken für das Wissensmanagement in Lern- und Kooperationsprozessen*. Göttingen: Hogrefe, S. 3-12.
- Marquardt, Nicolai; Nacenta, Miguel A.; Young, James E.; Carpendale, Sheelagh; Greenberg, Saul & Sharlin, Ehud (2009).** The Haptic Tabletop Puck: tactile feedback for interactive tabletops. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*, S. 85-92. New York: ACM. Online: [<http://doi.acm.org/10.1145/1731903.1731922>].
- Marquardt, Nicolai; Kiemer, Johannes; Ledo, David; Boring, Sebastian & Greenberg, Saul (2011).** Designing User-, Hand-, and Handpart-Aware Tabletop Interactions with the TOUCHID Toolkit. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. Kobe, Japan. Online: [<http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2011-TouchID.ITS.pdf>].
- McAdam, Christopher & Brewster, Stephen (2009).** Distal tactile feedback for text entry on tabletop computers. In: *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology (BCS HCI '09)*. Cambridge, United Kingdom — September 01 - 05, 2009. Swinton: British Computer Society, S. 504-511. Online: [<http://portal.acm.org/citation.cfm?id=1671011.1671076>].

- McFadzean, Elspeth (1997).** Improving Group Productivity with Group Support Systems and Creative Problem Solving Techniques. In: *Creativity and Innovation Management* 6 (4), S. 218–225. Online: [<http://dx.doi.org/10.1111/1467-8691.00072>].
- McLeod, Poppy Lauretta (1992).** An Assessment of the Experimental Literature on Electronic Support of Group Work: Results of a Meta-Analysis. In: *Human-Computer Interaction* 7 (3), S. 257–280. Online: [http://dx.doi.org/10.1207/s15327051hci0703_1].
- Meixner, André (2008).** *cgTable: Ein interaktives Tabletop-System zur Unterstützung kreativer Gruppenarbeit.* Online: [<http://flashingcreativity.blogspot.com/2008/09/cgtable-ein-interaktives-tabletop.html>].
- Meyer, Tobias & Schmidt, Dominik (2010).** IdWristbands: IR-based user identification on multi-touch surfaces. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. Saarbrücken, Germany — November 07 - 10, 2010. New York: ACM, S. 277–278. Online: [<http://doi.acm.org/10.1145/1936652.1936714>].
- Microsoft (2008).** *Microsoft Surface Virtual Pressroom.* Online: [<http://www.microsoft.com/presspass/presskits/surfacecomputing/gallery.mspix>].
- Microsoft (2009).** *User Experience Guidelines. User Interaction and Design Guidelines for Creating Microsoft Surface Applications.* Online: [<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=38CC76F1-4A16-4C13-9740-C34DBB5C3012&%3Bdisplaylang=en>].
- Microsoft (2011).** *About Microsoft Surface.* Online: [<http://www.microsoft.com/surface/en/us/Press.aspx>].
- Mitsubishi Electronic Research Laboratories (2008).** *DiamondSpin.* Online: [<http://www.merl.com/projects/diamondspin/>].
- Montferrat, P.; Apostolopoulo, A.; Besacier, G.; Buisine, S.; Rey, G. & Vernier, F. (2007).** Brainstorming and Mind-Mapping on DiamondTouch: Creativity Unleashed. In: *2nd DiamondTouch Workshop, from Personal Computer to Multi-user Collaboration*. 5 June 2007, Amsterdam, the Netherlands. Online: [<http://www.limsi.fr/Individu/besacier/papers/06-montferrat-dtw.pdf>].

- Morris, Meredith Ringel; Ryall, Kathy; Shen, Chia; Forlines, Clifton & Vernier, Frédéric (2004).** Beyond "social protocols": multi-user coordination policies for co-located groupware. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW '04)*. Chicago, IL, USA — November 06 - 10, 2004. New York: ACM, S. 262–265. Online: [<http://doi.acm.org/10.1145/1031607.1031648>].
- Morris, Meredith Ringel; Paepcke, Andreas; Winograd, Terry & Stamberger, Jeannie (2006a).** TeamTag: exploring centralized versus replicated controls for co-located tabletop groupware. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06)*. Montreal, Canada — April 22 - 27, 2006. New York: ACM, S. 1273–1282. Online: [<http://doi.acm.org/10.1145/1124772.1124964>].
- Morris, Meredith Ringel; Huang, Anqi; Paepcke, Andreas; Winograd, Terry (2006b).** Cooperative gestures: multi-user gestural interactions for co-located groupware. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06)*. Montreal, Canada — April 22 - 27, 2006. New York: ACM, S. 1201–1210. Online: [<http://doi.acm.org/10.1145/1124772.1124952>].
- Morris, Meredith Ringel (2008).** Slideshow: Tabletop Computers. In: *IEEE Spectrum Online*. Online: [<http://spectrum.ieee.org/computing/hardware/slideshow-tabletop-computers/0>].
- MT4j.org (2011).** *MT4j Architecture*. Online: [<http://mt4j.org/mediawiki/index.php/Architecture>].
- Mullen, Brian; Johnson, Craig & Salas, Eduardo (1991).** Productivity Loss in Brainstorming Groups: A Meta-Analytic Integration. In: *Basic & Applied Social Psychology* 12 (1), S. 3–23.
- Müller-Tomfelde, Christian & Fjeld, Morten (2010).** Introduction: A Short History of Tabletop Research, Technologies, and Products. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. London: Springer, S. 1–24.
- Murphy, Mark (2011).** *CommonsWare Android Components (CWAC)*. Online: [<http://commonsware.com/cwac>].

- Myers, Brad A.; Stiel, Herb & Gargiuio, Robert (1998).** Collaboration using multiple PDAs connected to a PC. In: *Proceedings of the 1998 ACM conference on Computer supported cooperative work (CSCW '98)*. Seattle, WA, USA — November 14 - 18, 1998. New York: ACM, S. 285-294. Online: [<http://doi.acm.org/10.1145/289444.289503>].
- Nacenta, Miguel A.; Gutwin, Carl; Aliakseyeu, Dzmitry & Subramanian, Sriram (2009).** There and Back Again: Cross-Display Object Movement in Multi-Display Environments. In: *Human-Computer Interaction* 24 (1/2), S. 170-229. Online: [<http://dx.doi.org/10.1080/07370020902819882>].
- Nacenta, Miguel A.; Pinelle, David; Gutwin, Carl & Mandryk, Regan (2010).** Individual and Group Support in Tabletop Interaction Techniques. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. London: Springer, S. 303-333.
- Nagel, Till; Pschetz, Larissa; Stefaner, Moritz; Halkia, Matina & Müller, Boris (2009).** mæve – An Interactive Tabletop Installation for Exploring Background Information in Exhibitions. In: *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*. (Proceedings, 3), S. 483-491. Online: [http://dx.doi.org/10.1007/978-3-642-02580-8_53].
- NUI Group Authors (2009).** *Multitouch Technologies*. Online: [http://nuicode.com/attachments/download/115/Multi-Touch_Technologies_v1.01.pdf].
- Nunamaker, J. F.; Dennis, Alan R.; Valacich, Joseph S.; Vogel, Douglas; George, Joey F. (1991).** Electronic Meeting Systems to support Group Work. In: *Communications of the ACM* 34 (7), S. 40-61. Online: [<http://doi.acm.org/10.1145/105783.105793>].
- O'Hara, Kenton (2010).** Interactivity and non-interactivity on tabletops. In: *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. Atlanta, GA, USA — April 10 - 15, 2010. New York: ACM, S. 2611-2614. Online: [<http://doi.acm.org/10.1145/1753326.1753723>].
- Offner, Anne K.; Kramer, Thomas J. & Winter, Joel P. (1996).** The Effects of Facilitation, Recording, and Pauses on Group Brainstorming. In: *Small Group Research* 27 (2), S. 283-298. Online: [<http://dx.doi.org/10.1177/1046496496272005>].

- Oppl, Stefan & Stary, Christian (2009).** Tabletop concept mapping. In: *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI '09)*. Regent, United Kingdom — February 16 - 18, 2009. New York: ACM, S. 275–282. Online: [<http://doi.acm.org/10.1145/1517664.1517721>].
- Osborn, A. F. (1953).** *Applied Imagination: Principles and procedures of creative thinking*. New York: Charles Scribner's Sons.
- Paivio, Allan (1979).** *Imagery and verbal processes*. Hillsdale: Erlbaum.
- Pal, Sujit (2006).** *Java Data Structure: A Generic Tree*. Salmon Run. Online: [<http://sujitpal.blogspot.com/2006/05/java-data-structure-generic-tree.html>].
- Parnes, Sidney J.; Meadow, Arnold (1959).** Effects of "brainstorming" instructions on creative problem solving by trained and untrained subjects. In: *Journal of Educational Psychology* 50 (4), S. 171–176.
- Paulus, Paul B.; Dzindolet, Mary T. (1993).** Social Influence Processes in Group Brainstorming. In: *Journal of Personality & Social Psychology* 64 (4), S. 575–586.
- Paulus, Paul B. & Yang Huei-Chuan (2000).** Idea generation in groups: A basis for creativity in organizations. In: *Organizational Behavior and Human Decision Processes* 82 (1), S. 76–87. Online: [<http://dx.doi.org/10.1006/obhd.2000.2888>].
- Poupyrev, Ivan & Maruyama, Shigeaki (2003).** Tactile interfaces for small touch screens. In: *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*. Vancouver, BC, Canada — November 02 - 05, 2003. New York: ACM, S. 217–220. Online: [<http://doi.acm.org/10.1145/964696.964721>].
- Prante, Thorsten; Magerkurth, Carsten & Streit, Norbert (2002).** Developing CSCW tools for idea finding -: empirical results and implications for design. In: *Proceedings of the 2002 ACM conference on Computer supported cooperative work (CSCW '02)*. New Orleans, LA, USA — November 16 - 20, 2002. New York: ACM, S. 106–115. Online: [<http://doi.acm.org/10.1145/587078.587094>].
- Prante, Thorsten; Streit, Norbert; Tandler, Peter (2004).** Roomware: Computers Disappear and Interaction Evolves. In: *Computer* 37 (12), S. 47–54. Online: [<http://doi.ieeecomputersociety.org/10.1109/MC.2004.255>].

- Reinig, Bruce A. & Shin, Bongsik (2002).** The Dynamic Effects of Group Support Systems on Group Meetings. In: *Journal of Management Information Systems* 19 (2), S. 303–325. Online: [http://layoftheland.net/archive/web/mis-575/course_docs/topic_6/Reinig.Shin.GSS.pdf].
- Rekimoto, Jun (1998).** A multiple device approach for supporting whiteboard-based interactions. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '98). Los Angeles, CA, USA — April 18 - 23, 1998. New York: ACM/Addison-Wesley Publishing Co., S. 344–351. Online: [<http://dx.doi.org/10.1145/274644.274692>].
- Rekimoto, Jun (2002).** SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* (CHI '02). Minneapolis, MN, USA — April 20 - 25, 2002. New York: ACM, S. 113–120. Online: [<http://doi.acm.org/10.1145/503376.503397>].
- Remy, Christian; Weiss, Malte; Ziefle, Martina & Borchers, Jan (2010).** A Pattern Language for Interactive Tabletops in Collaborative Workspaces. In: *Proceedings of the 15th European Conference on Pattern Languages of Programs* (EuroPLOP '10). Irsee Monastery, Germany, July 2010. Online: [<http://www.humtec.rwth-aachen.de/files/46.5.pdf>].
- Richter, Kai; Nichols, Jeffrey; Gajos, Krzysztof & Seffah, Ahmed (2006).** The Many Faces of Consistency in Cross-Platform Design. In: *CHI '06 extended abstracts on Human factors in computing systems* (CHI EA '06). Montreal, Canada — April 22 - 27, 2006. New York: ACM, S. 1639–1642. Online: [<http://doi.acm.org/10.1145/1125451.1125751>].
- Rogers, Yvonne & Lindley, Siân E. (2004).** Collaborating around vertical and horizontal large displays: Which way is best? In: *Interacting With Computers* 16 (6). Online: [http://www.informatics.sussex.ac.uk/research/groups/interact/publications/Rogers_final.pdf].
- Roth, Tim (2008).** *DSI - Diffused Surface Illumination*. Online: [<http://iad.projects.zhdk.ch/multitouch/?p=90>].

- Roth, Volker; Schmidt, Philipp & Güldenring, Benjamin (2010).** The IR ring: authenticating users' touches on a multi-touch display. In: *Proceedings of the 23rd annual ACM symposium on User interface software and technology* (UIST '10). New York, NY, USA — October 03 - 06, 2010. New York: ACM, S. 259–262. Online: [<http://doi.acm.org/10.1145/1866029.1866071>].
- Royce, Winston W. (1987).** Managing the development of large software systems: concepts and techniques. In: *Proceedings of the 9th international conference on Software Engineering* (ICSE '87). Los Alamitos: IEEE Computer Society Press, S. 328–338. Online: [<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>]
- Ryall, Kathy; Forlines, Clifton; Shen, Chia & Morris, Meredith Ringel (2004).** Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (CSCW '04). Chicago, IL, USA — November 06 - 10, 2004. New York: ACM, S. 284–293. Online: [<http://doi.acm.org/10.1145/1031607.1031654>].
- Ryall, Kathy; Forlines, Clifton; Shen, Chia; Morris, Meredith Ringel & Everitt, Katherine (2006a).** Experiences with and Observations of Direct-Touch Tabletops. In: *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems* (TABLETOP '06). Adelaide, Australia January 05-January 07. S. 89–96. Online: [<http://doi.ieeecomputersociety.org/10.1109/TABLETOP.2006.12>].
- Ryall, Kathy; Esenther, Alan; Forlines, Clifton; Shaer, Orit; Shipman, Sam; Morris, Meredith Ringel; Everitt, Katherine & Vernier, Frédéric D. (2006b).** Identity-Differentiating Widgets for Multiuser Interactive Surfaces. In: *IEEE Computer Graphics and Applications* 26 (5), S. 56–64. Online: [http://research.microsoft.com/en-us/um/people/merrie/papers/ryall_morris_ieee_cga.pdf].
- Saffer, Dan (2009).** *Designing gestural interfaces*. 1. Auflage. Sebastopol: O'Reilly.
- Sakurai, Satoshi; Kitamura, Yoshifumi; Subramanian, Sriram & Kishino, Fumio (2009).** A visibility control system for collaborative digital table. In: *Personal and Ubiquitous Computing* 13 (8), S. 619–632. Online: [<http://dx.doi.org/10.1007/s00779-009-0243-6>].
- Samsung (2011).** *Samsung SUR40 for Microsoft® Surface®*. Online: [<http://www.samsunglfd.com/product/spec.do?modelCd=SUR40>].

- Schmidt, Dominik; Chong, Ming Ki & Gellersen, Hans (2010a).** HandsDown: hand-contour-based user identification for interactive surfaces. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (NordCHI '10). New York: ACM, S. 432–441. Online: [<http://doi.acm.org/10.1145/1868914.1868964>].
- Schmidt, Dominik; Chong, Ming Ki & Gellersen, Hans (2010b).** IdLenses: Dynamic Personal Areas on Shared Surfaces. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '10). Saarbrücken, Germany — November 07 - 10, 2010. New York: ACM. Online: [<http://www.dfki.de/its2010/papers/sp112.html>].
- Scott, Stacey D.; Grant, Karen D. & Mandryk, Regan L. (2003).** System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In: *European Conference Computer-Supported Cooperative Work* (ECSCW 2003). September 14, 2003. Helsinki, Finland. Online: [<http://ilpubs.stanford.edu:8090/612/>].
- Scott, Stacey D.; Carpendale, Sheelagh & Inkpen, Kori M. (2004).** Territoriality in collaborative tabletop workspaces. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (CSCW '04). Chicago, IL, USA — November 06 - 10, 2004. New York: ACM, S. 294–303. Online: [<http://doi.acm.org/10.1145/1031607.1031655>].
- Scott, Stacey D. & Carpendale, Sheelagh (2010).** Theory of Tabletop Territoriality. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays* (*Human-Computer Interaction Series*). London: Springer, S. 357–385.
- Schöning, Johannes; Brandl, Peter; Daiber, Florian; Echtler, Florian; Hilgiges, Otmar; Hook, Jonathan; Löchtfeld, Markus; Motamedi, Nima; Muller, Laurence; Olivier, Patrick; Roth, Tim; Zadow, Ulrich von (2008).** *Multi-touch surfaces: A technical guide*. Technical Report TUM-10833. Technische Universität München. Online: [<http://campar.in.tum.de/pub/schoening2008multitouch/schoening2008multitouch.pdf>].
- Schöning, Johannes; Rohs, Michael & Krüger, Antonio (2008).** Spatial Authentication on Large Interactive Multi-Touch Surfaces. In: *Adjunct Proceedings of the Third IEEE International Workshop on Tabletops and Interactive Surfaces* (Tabletop 2008). Amsterdam, The Netherlands October 1-3, 2008. Online: [<http://www.medien.ifi.lmu.de/forschung/publikationen/detail?pub=schoening2008SpatialAuthentication>].

- Schöning, Johannes; Hook, Jonathan; Bartindale, Tom; Schmidt, Dominik; Oliver, Patrick; Echtler, Florian; Motamedi, Nima; Brandl, Peter & Zadow, Ulrich von (2010).** Building Interactive Multi-Touch Surfaces. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. S. 27-49. London: Springer
- Shaer, Orit; Kol, Guy; Strait, Megan; Fan, Chloe; Grevet, Catherine & Elfenbein, Sarah (2010).** G-nome surfer: a tabletop interface for collaborative exploration of genomic data. In: *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. Atlanta, GA, USA — April 10 - 15, 2010. New York: ACM, S. 1427-1436. Online: [<http://doi.acm.org/10.1145/1753326.1753539>].
- Shen, Chia; Ryall, Kathy; Forlines, Clifton; Esenther, Alan; Vernier, Frédéric D.; Everitt, Katherine; Wu, Mike; Wigdor, Daniel; Morris, Meredith Ringel; Hancock, Mark & Tse, Edward (2006).** Informing the Design of Direct-Touch Tabletops. In: *IEEE Computer Graphics and Applications* 26 (5), S. 36-46. Online: [http://katherineeveritt.com/papers/shen_morris_ieee_cga.pdf].
- Shih, Patrick C.; Nguyen, David H.; Hirano, Sen H.; Redmiles, David F. & Hayes, Gillian R. (2009).** GroupMind: supporting idea generation through a collaborative mind-mapping tool. In: *Proceedings of the ACM 2009 international conference on Supporting group work (GROUP '09)*. Sanibel Island, FL, USA — May 10 - 13, 2009. New York: ACM, S. 139-148. Online: [<http://doi.acm.org/10.1145/1531674.1531696>].
- Shoemaker, Garth B. D. & Inkpen, Kori (2001).** Single display privacyware: augmenting public displays with private information. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '01)*. Seattle, WA, USA — March 31 - April 05, 2001. New York: ACM, S. 522-529. Online: [<http://doi.acm.org/10.1145/365024.365349>].
- Sjurts, Insa (2011).** Smartphone. In: *Gabler Wirtschaftslexikon*. Online: [<http://wirtschaftslexikon.gabler.de/Archiv/569824/smartphone-vi.html>].
- Stanton, D.; Neale, H.R (2003).** The effects of multiple mice on children's talk and interaction. In: *Journal of Computer Assisted Learning* 19 (2), S. 229-238. Online: [<http://dx.doi.org/10.1046/j.0266-4909.2003.00023.x>].

- Steimle, Jürgen; Khalilbeigi, Mohammadreza; Mühlhäuser, Max & Hollan, James D. (2010).** Physical and Digital Media Usage Patterns on Interactive Tabletop Surfaces. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. Saarbrücken, Germany — November 07 - 10, 2010. New York: ACM. Online: [<http://www.dfki.de/its2010/papers/fp208.html>].
- Stewart, Jason; Bederson, Benjamin B. & Druin, Allison (1999).** Single display groupware: a model for co-present collaboration. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '99)*. Pittsburgh, PA, USA — May 15 - 20, 1999. New York: ACM, S. 286–293. Online: [<http://doi.acm.org/10.1145/302979.303064>].
- Streitz, Norbert A.; Rexroth, Petra; Holmer, Torsten (1997).** Does "roomware" matter ? Investigating the role of personal and public information devices and their combination in meeting room collaboration. In: *Proceedings of the European Conference on Computer-Supported Cooperative Work (E-CSCW'97)*. September 7-11, 1997, Lancaster, UK. Dordrecht: Kluwer Academic Publishers. [<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.917>]
- Streitz, Norbert A.; Geißler, Jörg; Holmer, Torsten; Konomi, Shin'ichi; Müller-Tomfelde, Christian & Reischl, Wolfgang et al. (1999).** i-LAND: an interactive landscape for creativity and innovation. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '99)*. Pittsburgh, PA, USA — May 15 - 20, 1999. New York: ACM, S. 120–127. Online: [<http://doi.acm.org/10.1145/302979.303010>].
- Streitz, Norbert A. (2010).** Foreword: From Furniture to Interaction Substrate. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. London: Springer, S. v–viii.
- Streng, Sara; Stegmann, Karsten; Boring, Sebastian; Böhm, Sonja; Fischer, Frank; Hussmann, Heinrich (2010).** Measuring effects of private and shared displays in small-group knowledge sharing processes. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. New York: ACM, S. 789–792. Online: [<http://doi.acm.org/10.1145/1868914.1869030>].
- Svantesson, Ingemar (2001).** *Mind mapping und Gedächtnistraining. Übersichtlich strukturieren ; kreativ arbeiten ; sich mehr merken*. Offenbach: GABAL.

- Tandler, Peter; Prante, Thorsten; Müller-Tomfelde, Christian; Streit, Norbert & Steinmetz, Ralf (2001).** Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology* (UIST '01). Orlando, FL, USA — November 11 - 14, 2001. New York: ACM, S. 11-20. Online: [<http://doi.acm.org/10.1145/502348.502351>].
- Tang, John C. (1991).** Findings from observational studies of collaborative work. In: *International Journal of Man-Machine Studies* 34 (2), S. 143-160. Online: [[http://dx.doi.org/10.1016/0020-7373\(91\)90039-A](http://dx.doi.org/10.1016/0020-7373(91)90039-A)].
- Tassoul, Marc & Buijs, Jan (2007).** Clustering: An Essential Step from Diverging to Converging. In: *Creativity and Innovation Management* 16 (1), S. 16-26. Online: [<http://vir.liu.se/~TDDC62/project/Konceptframtagning/clustering.pdf>].
- Taylor, D.W., Berry, P.C. & Block, C. H. (1958).** Does group participation when using brainstorming facilitate or inhibit creative thinking? In: *Administrative Science Quarterly* 3 (1), S. 22-47.
- Terrenghi, Lucia; Kirk, David; Richter, Hendrik; Krämer, Sebastian; Hilliges, Otmar & Butz, Andreas (2008).** Physical handles at the interactive surface: exploring tangibility and its benefits. In: *Proceedings of the working conference on Advanced visual interfaces* (AVI '08), S. 138-145. Online: [<http://doi.acm.org/10.1145/1385569.1385593>].
- Terrenghi, Lucia; Quigley, Aaron & Dix, Alan (2009).** A taxonomy for and analysis of multi-person-display ecosystems. In: *Personal and Ubiquitous Computing* 13 (8), S. 583-598. Online: [<http://dx.doi.org/10.1007/s00779-009-0244-5>].
- Thuy, Luu Gia (2011).** *Simple Android and Java Bluetooth Application*. Online: [<http://luugiathuy.com/2011/02/android-java-bluetooth/>]
- Tuio.org (2011).** *TUIO*. Online: [<http://www.tuio.org>].
- Tzounopoulos, Kostas (2007).** *Microsoft Surface Computing and Zune*. Zune-Online.com. Online: [<http://www.zune-online.com/news/zune/microsoft-surface-computing-and-zune.html>].
- Ullmer, Brygg & Ishii, Hiroshi (1997).** The metaDESK: models and prototypes for tangible user interfaces. In: *Proceedings of the 10th annual ACM symposium on User interface software and technology* (UIST '97). Banff, AB, Canada — October 14 - 17, 1997. New York: ACM, S. 223-232. Online: [<http://doi.acm.org/10.1145/263407.263551>].

- Ullmer, Brygg (2002).** *Tangible interfaces for manipulating aggregates of digital information*. Dissertation. Cambridge: Massachusetts Institute of Technology. Online: [<http://alumni.media.mit.edu/~ullmer/thesis/full.pdf>].
- Usability in Germany (2011).** *Glossar: Affordance, Affordanz*. Online: [<http://www.usability-in-germany.de/definition/affordance-affordanz>].
- VanGundy, Arthur B. (1984).** Brainwriting for new product ideas: An alternative to brainstorming. In: *Journal of Consumer Marketing* 1 (2), S. 67–74. Online: [<http://dx.doi.org/10.1108/eboo8097>]
- Van Boeijen, Annemiek & Daalhuizen, Jaap (2010).** Creativity techniques. In: Van Boeijen, Annemiek & Daalhuizen, Jaap (Hrsg.). *Delft Design Guide*. Delft University of Technology. Online: [http://www.wikid.eu/index.php/Creativity_techniques].
- Wallace, James R. & Scott, Stacey D. (2008).** Contextual Design Considerations for Co-located, Collaborative Tables. In: *Third IEEE International Workshop on Tabletops and Interactive Surfaces* (Tabletop 2008). Amsterdam, The Netherlands October 1–3, 2008. S. 57–64. Online: [http://www.eng.uwaterloo.ca/~s9scott/wiki/uploads/Main/wallace_tabletop2008.pdf].
- Wallace, James R.; Scott, Stacey D.; Stutz, Taryn; Enns, Tricia & Inkpen, Kori (2009).** Investigating teamwork and taskwork in single- and multi-display groupware systems. In: *Personal and Ubiquitous Computing* 13 (8), S. 569–581. Online: [<http://dx.doi.org/10.1007/s00779-009-0241-8>].
- Wang, Xin; Ghanam, Yaser & Maurer, Frank (2008).** From Desktop to Tabletop: A Migration of User Interface Design for Agile Planner. In: *Engineering Interactive Systems 2008 - The 2nd Conference on Human Centered Software Engineering* (EIS/HCSE2008). Italy. Online: [<http://ase.cpsc.ucalgary.ca/ase/uploads/Publications/eics.pdf>].
- Wang, Feng & Ren, Xiangshi (2009).** Empirical evaluation for finger input properties in multi-touch interaction. In: *Proceedings of the 27th international conference on Human factors in computing systems* (CHI '09), S. 1063–1072. New York: ACM. Online: [<http://doi.acm.org/10.1145/1518701.1518864>].
- Weiser, Mark (1991).** The Computer for the 21st Century. In: *Scientific American (Special Issue on Communications, Computers, and Network)* 3 (3), S. 94–104. Online: [http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf].

- Weiss, Malte; Hollan, James D. & Borchers, Jan (2010).** Augmenting Interactive Tabletops with Translucent Tangible Controls. In: Christian Müller-Tomfelde (Hrsg.). *Tabletops - Horizontal Interactive Displays (Human-Computer Interaction Series)*. London: Springer, S. 149–170.
- Wellner, Pierre David (1993).** *Interacting with paper on the DigitalDesk*. Dissertation. University of Cambridge. Online : [http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/PhD-Thesis%20%28Wellner%201993%29.pdf].
- Wigdor, Daniel & Wixon, Dennis (2011).** *Brave NUI world. Designing natural user interfaces for touch and gesture*. Burlington: Morgan Kaufmann.
- Wobbrock, Jacob O.; Morris, Meredith Ringel & Wilson, Andrew D. (2009).** User-defined gestures for surface computing. In: *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09)*, S. 1083–1092. Online: [http://doi.acm.org/10.1145/1518701.1518866].
- Wu, Mike & Balakrishnan, Ravin (2003).** Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*. Vancouver, BC, Canada — November 02 - 05, 2003. New York: ACM, S. 193–202. Online: [http://doi.acm.org/10.1145/964696.964718].
- Xerox Corporation (2011).** *Research & Development: DigitalDesk to CamWorks*. Online: [http://www.xrce.xerox.com/Research-Development/Historical-projects/DigitalDesk-to-CamWorks].