



MASTERARBEIT

Analyse von Varianten der Mensch-Computer-Kommunikation

Ausgeführt am
Institut für Gestaltungs- und Wirkungsforschung
der Technischen Universität Wien

unter der Leitung von
Univ.-Prof. Dr. Ina Wagner

unter Mitbetreuung von
Univ.-Ass. Dr. Thomas Psik

durch
JOHANNES SCHMIDMAYR, BSc
Matr.-Nr. 0425942

Koppstrasse 42/11
A-1160 Wien

Wien, am 21. Mai 2007

Kurzfassung

Die Konzeption und Gestaltung anwenderfreundlicher Benutzerschnittstellen ist eine bedeutende Herausforderung und stellt permanent neue Anforderungen an Designer, Techniker und Wissenschaftler. Dieser Prozess erfordert detailliertes Spezialwissen und bindet Expertisen aus verschiedenen Fachrichtungen mit ein. Interaktionsdesign wird dabei als Teilgebiet der wissenschaftlichen Auseinandersetzung mit der Mensch-Computer-Kommunikation zu einer wesentlichen Tätigkeit.

Die vorliegende Arbeit beschäftigt sich mit der Beschreibung und Illustration jener Modelle und Konzepte, die eine Ausgangsbasis für die Gestaltung der Interaktion in multimedialen Systemen darstellen. Entscheidend ist, dass sehr viele unterschiedlich Ansätze und Möglichkeiten der Kommunikation zwischen Mensch und Computer bestehen. Diese werden innerhalb dieser Arbeit als so genannte *Interaktionsparadigmen* bezeichnet. Sie legen Art und Weise bzw. Stil der Interaktion fest und sind je nach Anwendungsfeld unterschiedlich.

An Hand der Theorien von Wissenschaftlern und anerkannten Experten auf dem Gebiet der *HCI* wird eine Definition des Begriffes Interaktionsparadigma gegeben. Die gewonnenen Erkenntnisse bilden in weiterer Folge die Grundlage für die praktische Umsetzung eines Beispiels im Rahmen eines Design- und Implementierungsprozesses. Hierzu wird das von Jef Raskin kreierte Modell der *Zooming Interfaces* herangezogen. Zur Interaktion werden die Konzepte der *Gesture Based* bzw. *Tangible Interaction* eingesetzt, was die Konstruktion eines interaktiven Setups in Form eines kamerabasierenden Trackingsystems erforderte.

In den folgenden Kapiteln werden die Grundlagen zur Themenstellung genau erörtert und sämtliche Teilbereiche der praktischen Umsetzung vorgestellt. Das Hauptaugenmerk liegt dabei vor allem auf der Präsentation und Diskussion geplanter Vorgehensweisen, gefundener Lösungen und getroffener Schlussfolgerungen.

Abstract

The conception and design of interactive User Interfaces is an important challenge and permanently producing new requirements for designers, engineers and scientists. Special knowledge is needed for this process and expertise from different fields is involved. Interaction Design has become an essential task within Human-Computer Communication.

This thesis deals with different models and concepts that are fundamental for designing interaction in multimedia-systems. There are many possibilities for communication between humans and machines. Within this thesis they are called *Interaction Paradigms*. They affect the character of interactive systems.

A definition for Interaction Paradigm will be given according to the theories of scientists and experts in *HCI*. These facts are important for the design process and the following implementation. Therefore, Jef Raskin's concept of the *Zooming Interfaces* will be realized. For interaction, the principles of *Gesture Based* and *Tangible Interaction* are implemented in combination with an optical tracking system.

Within the following chapters different Interaction Paradigms will be presented and discussed. Additionally, the solutions and results of the process of implementation are shown.

Danksagung

Ich möchte mich bei all jenen bedanken, die direkt oder indirekt zur Entstehung dieser Arbeit in Form fachlicher oder anderweitiger Unterstützung beigetragen haben.

Besonderer Dank gebührt dabei meinen Eltern, die mir durch ihre finanzielle und persönliche Unterstützung dieses Studium ermöglicht haben.

Weiters bedanke ich mich bei Frau Univ.-Prof. Dr. Ina Wagner sowie Herrn Univ.-Ass. Dr. Thomas Psik für die Betreuung meiner Arbeit. Ihre zahlreichen wissenschaftlichen Ratschläge waren für mich sehr wertvoll und haben stets zur Verbesserung dieser Arbeit beigetragen.

Inhaltsverzeichnis

Kurzfassung	i
Abstract	ii
Danksagung	iii
1 Einleitung	1
1.1 Motivation und Zielsetzung	3
1.2 Interaktionsparadigma: Versuch einer Begriffsdefinition . . .	3
2 Varianten der Mensch-Computer-Kommunikation	5
2.1 Interaktionstheorien und Systemparadigmen nach Herzeg . .	6
2.1.1 Systemparadigmen	6
2.1.2 Varianten von SP I: Kommunikationspartner	8
2.1.3 Varianten von SP II: Handlungsräume	10
2.1.4 Varianten von SP III: Mediale Systeme	13
2.1.5 Zusammenfassung	15
2.2 Theorien, Modelle und Interaktionsstile nach Shneiderman . .	17
2.2.1 Interaktionstheorien: Object-Action Interface Model .	17
2.2.2 Varianten von IS I: Direkt manipulative Systeme . . .	19
2.2.3 Varianten von IS II: Menüauswahl, Formularfelder . .	21
2.2.4 Varianten von IS III: Sprachen	23
2.2.5 Zusammenfassung	25
2.3 Interfacemetaphern nach Jenny Preece	26
2.3.1 Grundzüge von konzeptionellen Modellen	27
2.3.2 Varianten von Interaktionsparadigmen	30
2.3.3 Zusammenfassung	33
2.4 Interfaceparadigmen nach Jef Raskin	34
2.4.1 Interaktionsanalyse im Kontext der Wahrnehmung . .	35
2.4.2 Intuitive und Natürliche Interaktion	37
2.4.3 Techniques in Humane Interfaces	38
2.4.4 Zusammenfassung	40
2.5 Die Interaktionsparadigmen von Jakob Nielsen	41
2.5.1 Das Ende von WYSIWYG	41

2.6	Varianten und Formen der direkten Interaktion	43
2.6.1	Graphical User Interfaces – GUIs	43
2.6.2	Pointing Devices	44
2.6.3	Tangible Interaction	46
2.6.4	Sound and Speech Interaction	47
2.6.5	Gesture Based Interaction	49
2.7	Zusammenfassung und Analyse	50
3	Designkonzept	54
3.1	Ausgangsposition	54
3.2	Konzeptionierung des interaktiven Systems	56
3.2.1	Setup	57
3.2.2	Object-Tracking – Farbmarker	60
3.3	Entwurf und Skizzierung	62
3.3.1	Gesture Based Interaction	63
3.3.2	Tangible Interaction	66
4	Implementierung	69
4.1	Systemeinsatz und Entwicklungsumgebung	69
4.1.1	Programmiersprache Lingo	70
4.1.2	Zusatzmodul - TrackThemColors	70
4.2	Grundkonzept der Bewegungserkennung	71
4.2.1	Problemstellungen bei der Aufgabendurchführung	73
4.3	Aufbau der Applikation	74
4.3.1	Gesture Based Interaction im Zoom Space	77
4.3.2	Tangible Interaction im Zoom Space	81
4.3.3	Problemstellungen bei der Aufgabendurchführung	84
4.4	Vergleich: Tangible vs. Gesture Based Interaction	86
5	Schlussbemerkungen	88
5.1	Ausblick für die Zukunft	89
A	Inhalt der CD-ROM	91
A.1	Kurzfassung / Abstract	91
A.2	Masterarbeit	91
A.3	Software	91
A.4	Videos	91
	Literaturverzeichnis	92

Abbildungsverzeichnis

2.1	Systemparadigma Kommunikationspartner	8
2.2	Beispiel für einen Conversational Agent	9
2.3	Systemparadigma Handlungsräume	10
2.4	Beispiel für Augmented Reality	12
2.5	Systemparadigma Mediale Systeme	13
2.6	Object-Action Interface Model	18
2.7	Direkt manipulatives System: Adobe Photoshop®	20
2.8	Varianten von Menüsystemen	21
2.9	Interaktion mittels Befehlssprachen	24
2.10	Beispiel für ein Desktop-System	28
2.11	Implementierung des ZIP I	39
2.12	Implementierung des ZIP II	39
2.13	Beispiele für Pointing Devices	45
2.14	Tangible Interaction am Beispiel des ColorTable	47
2.15	Gesture Based Interaction	49
3.1	Interaktives System – Gesamtsetup	58
3.2	Interaktives System – Object-Tracking	59
3.3	Interaktives System – Informationsmonitor	60
3.4	Object Tracking – Farbmarker Gesture Based Interaction	61
3.5	Object Tracking – Farbmarker Tangible Interaction	61
3.6	Erster Entwurf der grafischen Oberfläche	63
3.7	GBI – Grafischer Entwurf für die Aktion des Zooming	63
3.8	GBI – Grafischer Entwurf für die vertikale Navigation	64
3.9	GBI – Grafischer Entwurf für die horizontale Navigation	64
3.10	GBI – Grafischer Entwurf für die Rotation	65
3.11	GBI – Grafischer Entwurf für die Auswahl eines Elements	65
3.12	TI – Grafischer Entwurf für die Aktion des Zooming	67
3.13	TI – Grafischer Entwurf für die Navigation	67
3.14	TI – Grafischer Entwurf für die Selektierung	68
4.1	Object Tracking – Initialisierung der Farbwerte	71
4.2	Applikation – Aufbau des sekundären Monitors	75

4.3	Applikation – Grafische Gestaltung der primären Anzeige . .	76
4.4	Piktogramme – Darstellung durch Symbole	76
4.5	GBI – Übertragung der Trackingdaten	78
4.6	GBI – Praktische Umsetzung des Zooming	78
4.7	GBI – Navigation auf der grafischen Oberfläche	79
4.8	GBI – Rotation des Arbeitsbereiches	80
4.9	GBI – Selektierung durch das Auslösen eines Impulses	80
4.10	Applikationsbereich	81
4.11	TI – Zooming	82
4.12	TI – Navigation und Selektierung	83
4.13	Modifizierte Farbmarker	84
4.14	Entstehung des Öffnen-Schließen-Problems	85

Kapitel 1

Einleitung

Der allgegenwärtige Einsatz von Computersystemen stellt zunehmend neue Anforderungen im Bereich der Konzeption, Konstruktion, Gestaltung und Realisierung anwenderfreundlicher Benutzerschnittstellen. Durch intensive Forschung und Entwicklung wird das Aufgabengebiet von Computern mehr und mehr erweitert. Die vermehrte Integration von multimedialen Inhalten in viele Utensilien des alltäglichen Lebens, wird einerseits von Entwicklungen im technischen Bereich und andererseits von Veränderungen im Konsumverhalten der Menschen getragen. In der Gesellschaft scheint es akzeptiert, dass Computer Bestandteil des alltäglichen Lebens sind und in quasi allen Lebenssituationen auftauchen. Dadurch ergeben sich natürlich völlig neue Formen der Kommunikation zwischen Mensch und Computer, was wiederum permanent neue Herausforderungen an Designer, Informatiker, Techniker und Wissenschaftler stellt. Sie sind es, die durch ihre Arbeit die Ausgangspunkte und Konzepte für die Kreation neuer, intuitiver und überzeugender Interaktion schaffen.

Interaktionsdesign wird als Teilgebiet der wissenschaftlichen Auseinandersetzung mit der Mensch-Computer-Kommunikation zu einer wesentlichen Tätigkeit. Es handelt sich dabei um eine intensive und umfangreiche Tätigkeit, die Expertisen in den verschiedensten Fachgebieten erfordert. Als multifunktionaler Designbereich, vereint es Elemente aus Sozialwissenschaft, Design und Technik. Die Gestaltung interaktiver Systeme wird dadurch zu einer vielseitigen Problemstellung, die aus verschiedenen fachlichen Perspektiven gesehen werden kann. Faktoren wie Technik und Informatik, Psychologie und Ästhetik sowie kulturelle und wirtschaftliche Aspekte müssen aufeinander abgestimmt werden, um ein passendes Zusammenwirken zu erreichen. Zudem gilt es die Ansätze und Theorien einer ständigen Kontrolle und Weiterentwicklung zu unterziehen, um zusätzlich in neue wissenschaftliche Bereiche vorzudringen.

Um zumindest eine gewisse Eingrenzung vornehmen zu können, wird die Begriffsthematik innerhalb des hier vorherrschenden Zusammenhanges, vor-

rangig aus software- und medientechnischer Sicht gesehen. Interaktionsdesign spielt in diesen Disziplinen eine entscheidende Rolle, in der Entwicklung und im Design von Modellen und Systemen.

Die vorliegende Arbeit beschäftigt sich nun mit der Darstellung und Konkretisierung der Modelle, Theorien und Metaphern, die quasi die Grundlage für die Konzeption und Gestaltung der Interaktion in multimediale Systemen darstellen. Man spricht in diesem Zusammenhang auch von Formen der Mensch-Computer-Interaktion (Englisch: *Human-Computer-Interaction, HCI*). Entscheidend ist, dass sehr viele unterschiedliche Ansätze und Möglichkeiten der Kommunikation zwischen Mensch und Computer bestehen, in weiterer Folge als so genannte *Interaktionsparadigmen* bezeichnet. Diese Paradigmen legen Art und Weise bzw. Stil der Interaktion fest und sind je nach Anwendungsfeld unterschiedlich. Für die Gestaltung der Benutzerschnittstelle bilden sie aber eine maßgebliche Grundlage.

In den ersten beiden Kapiteln wird versucht, an Hand konkreter Beispiele eine Definition für den Begriff des Interaktionsparadigmas zu finden. Zu diesem Zweck werden auf Basis einer intensiven Literaturrecherche Definitionen von Wissenschaftlern, Forschern und anerkannten Experten auf dem Gebiet der *HCI* präsentiert. Dadurch soll die Möglichkeit geboten werden, spezifische Grundcharakteristika zu erkennen und zu klassifizieren. In einer Analyse werden die Erkenntnisse des Forschungsteils noch einmal gegenüber gestellt – um daraus resultierend – die Grundlagen für den zweiten Abschnitt dieser Arbeit abzuleiten. Dieser beschäftigt sich mit der konkreten praktischen Umsetzung eines definierten Interaktionsparadigmas. Von der Festlegung des Designkonzeptes bis hin zur finalen Implementierung werden sämtliche Arbeitsschritte genau dokumentiert, um die Vorgehensweise strukturiert zu veranschaulichen. Nicht zuletzt spielen auch die persönlichen Erfahrungen eine wesentliche Rolle.

Der Aufbau der Arbeit wurde aus diesem Grund folgendermaßen gewählt:

- Kapitel 1: Einführung in die Problemstellung
- Kapitel 2: Definition und Präsentation von Interaktionsparadigmen an Hand konkreter Beispiele
- Kapitel 3: Formulierung einer Designidee auf Basis der Ergebnisse des Forschungsteiles
- Kapitel 4: Dokumentation der praktischen Umsetzung des Designkonzeptes
- Kapitel 5: Schlussbemerkungen, Evaluierung der Ergebnisse und Ausblick auf weiterführende Arbeit

Im angefügten Literaturverzeichnis finden sich verschiedene Publikationen, die zur Erstellung dieser Arbeit herangezogen wurde. Da es sich bei Interaktionsdesign um ein sehr weitläufiges Fachgebiet handelt, finden sich in

der Auflistung der verwendeten Literatur Werke aus den Bereichen Technik und Sozialwissenschaft und zu einem geringen Anteil aus der Psychologie. Auf der beigelegten CD-ROM sind – neben einer Kurzfassung und der Arbeit selbst – die erstellte Software, in Form einer ausführbaren Datei und Videos, die eine Dokumentation des Ergebnisses der praktischen Umsetzung zeigen, abgelegt.

1.1 Motivation und Zielsetzung

Ein entscheidender Grund für die Themenwahl ist das persönliche Interesse für das Gebiet des Interaktionsdesigns. Dieses wurde hervorgerufen durch verschiedene Lehrveranstaltungen, die im Rahmen des Masterstudiums Medieninformatik absolviert wurden, sowie durch die Mitarbeit in diversen Projekten zur Thematik, auf die hier aber nicht genauer eingegangen werden soll.

Durch die Verfassung einer entsprechenden Studienabschlussarbeit, sollte die Möglichkeit einer zusätzlichen Vertiefung in die Materie gegeben sein, um damit nicht zuletzt auch die Weichen für eine mögliche berufliche Tätigkeit in diesem Bereich zu legen.

Zu Beginn war es das erklärte Ziel, den Prozess des Interaktionsdesigns, wenn auch nur in Grundzügen, an Hand eines praktischen Beispiels zu zeigen. Auf Basis eines Forschungsteiles sollte ein geeignetes Fundament geschaffen werden damit – darauf aufbauend – eine weiterführende praktische Umsetzung durchgeführt werden kann. Hierzu wurde eine Analyse etablierter Interaktionstechniken, im Rahmen der Arbeit als Interaktionsparadigmen bezeichnet, als passend angesehen.

In den folgenden Kapiteln werden die Grundlagen zur Themenstellung genau erörtert, Vorgehensweisen, Lösungen und Schlussfolgerungen präsentiert und schlussendlich die Ergebnisse diskutiert und evaluiert.

1.2 Interaktionsparadigma: Versuch einer Begriffsdefinition

Die Möglichkeiten den Benutzer in das Aktionsfeld eines interaktiven Systems einzubinden, sind sehr vielfältig und haben sich in den letzten Jahren stark entwickelt und gewandelt. Die Wahl des passenden Interaktionsparadigmas ist ein fundamentales Element im gesamten Design- und Implementierungsprozess. Für eine genaue Definition des Begriffes Interaktionsparadigma ist eine konkrete Analyse notwendig, um Spezifikationen und Eigenschaften festzulegen. In dieser Formulierung manifestiert sich aber auch die Grundproblematik dieser Aufgabenstellung. Mehrere Definitionen existieren parallel und prinzipiell bestehen multiple Möglichkeiten um zu spezifizieren, wie Interaktion zwischen Mensch und Maschine an sich funktionieren kann.

Wurde für die Beschreibung der Thematik innerhalb dieser Arbeit die Bezeichnung Interaktionsparadigma gewählt, werden in der Literatur auch Begriffe wie Interaktionsmodell, -stil, -metapher, -theorie oder auch -strategie verwendet. Vorweggenommen sei gesagt, dass der Begriff des Interaktionsparadigmas (noch) nicht eindeutig zu beschreiben ist, da keine definitiven Festlegungen getroffen sind. Dies mag wohl auch daran liegen, dass es sich beim Interaktionsdesign an sich um eine Tätigkeit handelt, die eine Expertise verschiedener Wissensgebiete erfordert und deren Bedeutung in den letzten Jahren verstärkt zugenommen hat.

Für eine Unterscheidung und Einordnung von Interaktionsparadigmen ist es notwendig, gewisse grundsätzliche Merkmale zu definieren, an Hand derer eine Klassifizierung durchgeführt werden kann. Hierbei ist es in erster Linie essentiell, die Art und Weise von Ein- und Ausgabevorgängen zu untersuchen. Ausschlaggebend ist dabei, welche Möglichkeiten den Benutzern zur Verfügung stehen, um Inputs zu generieren und in welchem Zusammenhang diese Dateneingaben mit dem Output des Systems stehen. Die daraus resultierende Form der Kommunikation erlaubt es dem Benutzer, auf Informationseinheiten zuzugreifen und Parameter zu definieren bzw. zu steuern und bietet auch dem Systemdesigner die Möglichkeit, durch entsprechende Leitung und Aufforderung innerhalb des Programms, die interagierende Person zu entsprechenden Handlungen zu veranlassen. Dadurch kann sich ein bidirektionaler Datenfluss entwickeln, der wie ein Dialog zwischen Mensch und Maschine unter gewissen zeitlichen und räumlichen Gegebenheiten stattfindet. Die grundsätzlichen Kriterien, an Hand derer Interaktionsparadigmen innerhalb dieser Arbeit unterschieden werden, gestalten sich also wie folgt:

- Formen von Ein- und Ausgabevorgängen
- Zusammenhang zwischen Input und Output
- Varianten der Kommunikation
- Gestaltung von Dialog und Handlung
- Zeitliche und räumliche Gegebenheiten

Vorab lässt sich sagen, dass sich die Differenzierung zwischen den einzelnen Bereichen als sehr schwierig erweist, was primär an den vielen Überschneidungen zwischen diesen liegt. Um nun das Prinzip des Interaktionsparadigmas einigermaßen greifbar zu machen, ist es notwendig Beispiele anzuführen, mit deren Hilfe eine eindeutige Beschreibung getroffen werden kann. Zu diesem Zweck wurde eine intensive Recherche durchgeführt, um herauszufinden, wie der Begriff in der Literatur verwendet wird. Die Ergebnisse hierzu werden im nachfolgenden Kapitel präsentieren.

Kapitel 2

Interaktionsparadigmen als Varianten der Mensch-Computer-Kommunikation

Interaktionsdesign ist eine multidisziplinäre Designdisziplin, die sich im Spannungsfeld verschiedenster Wissenschaften befindet. Für die technischen Wissenschaften, allen voran die Informatik, aus deren Blickwinkel die Thematik innerhalb dieser Arbeit betrachtet wird, spielt das Gestalten von wertvoller, interessanter und überzeugender Information und Interaktion eine wesentliche Rolle. Art und Weise der Kommunikation zwischen System und Benutzer sind dabei wesentliche Aspekte.

Die verschiedenen Formen des Zusammenwirkens zwischen Mensch und Computer stellen die Grundlage für den folgenden Abschnitt dar, in dem versucht wird, die Bedeutung des Begriffes Interaktionsparadigma näher zu erläutern. Dabei ist es entscheidend abzugrenzen, was nun als Interaktionsparadigma bezeichnet werden kann, welche charakteristischen Eigenschaften ausschlaggebend und bezeichnend sind und darüber hinaus, welcher primäre Weg der Kommunikation im entsprechenden Interaktionsmodell verwirklicht ist.

Zu diesem Zwecke werden verschiedene Beispiele aus der Literatur angeführt. Dies soll dabei nicht nur eine Auflistung unterschiedlicher Interaktionsformen darstellen, sondern vor allem auch die Möglichkeit bieten, spezifische Grundcharakteristika der unterschiedlichen Interaktionskonzepte zu erfassen und zu klassifizieren.

Eine entscheidende Erkenntnis, die vorab festgehalten werden kann ist, dass für ein und den selben Sachverhalt von vielen Autoren zwar unterschiedliche Bezeichnungen verwendet werden, aber inhaltlich dennoch starke Kongruenzen auftreten.

2.1 Interaktionstheorien und Systemparadigmen nach Herzeg

A good design is better than you think (Rex Heftman)

Der deutsche Wissenschaftler Michael Herzeg ist Universitätsprofessor und Direktor des Institutes für multimediale und interaktive Systeme an der Universität Lübeck. Er setzt sich in seinen Publikationen vorrangig mit Mustern und Modellen der Softwareergonomie auseinander. Dabei knüpft er nahtlos an die Konzepte und Theorien des Systemdesigns in der Informatik und Interfacegestaltung an. Zusätzlich versucht er einen genauen Überblick über die Herausforderungen des Interaktionsdesigns zu geben, um nicht zuletzt auch aktuelle Systemkonzepte und deren Benutzerschnittstellen zu evaluieren. Die Verbindung zur Thematik ergibt sich bei Herzeg durch seine langjährige Tätigkeit in den Bereichen Mensch-Computer-Interaktion, Softwareergonomie, Interaktionsdesign und sicherheitskritische Mensch-Maschine-Systeme. Dies macht ihn zu einem anerkannten Forscher und Wissenschaftler auf dem Gebiet. Aus diesem Grunde werden u.a. seine Theorien und Thesen herangezogen, um eine Definition des Begriffes Interaktionsparadigma zu liefern.

2.1.1 Systemparadigmen: Aufgabenteilung zwischen Mensch und Computer

Um nun ein Verständnis für die Theorien von Herzeg zu entwickeln, ist es vor allem notwendig, gewisse Grundbegriffe seiner Sichtweise der Dinge zu spezifizieren. Vorrangig beschäftigt er sich in [24] mit den Ausprägungen von Mensch-Computer-Systemen und spricht dabei im Kontext von Interaktion im Konkreten von so genannten *Paradigmen multimedialer interaktiver Systeme oder kurz Systemparadigmen*.¹

Die Klassifikation und Deskription der einzelnen Systemparadigmen stützt sich dabei auf bestimmte Charakteristika und Grundprinzipien der *HCI*. Seiner Meinung nach ist es ein primäres Ziel der Computernutzung, eine passende und adäquate Aufgabenteilung zu finden zwischen Mensch und Maschine. Art und Weise dieser Arbeitsaufteilung, stellt dabei sozusagen die Grundlage für das entsprechende, zu verwirklichende Systemparadigma dar. Entscheidend sei, dass in diesem Wechselspiel gleichermaßen die Fähigkeiten von Mensch und Computer einfließen.

Im Folgenden findet sich hierzu eine Auflistung und Gegenüberstellung von Eigenschaften und potentiellen Aufgabengebieten, die eine Ausgangsbasis für das Design von interaktiven Systemen darstellen können.²

¹Siehe [24]: Herzeg Michael: *Interaktionsdesign – Gestaltung interaktiver und multimedialer Systeme*. Oldenbourg Wissenschaftsverlag GmbH, 2006: S. 41

²Siehe [24]: Herzeg, 2006: S. 42

Der Mensch

- *setzt Ziele,*
- *benutzt sein leistungsfähiges System zur Wahrnehmung von Situationen und Ergebnissen,*
- *definiert Teilprobleme und ihre Beziehungen zueinander,*
- *wählt geeignete Problempräsentationen,*
- *benutzt umfangreiches Allgemeinwissen und integriert Wissen aus verschiedenen Bereichen zur Problemlösung,*
- *baut auf vorhergehende Erfahrungen auf und löst Probleme durch Analogieschlüsse,*
- *kontrolliert die Teillösungen und fügt diese zu einer Gesamtlösung zusammen und*
- *führt komplexe Entscheidungen durch.*

Der Computer hingegen

- *übernimmt die Rolle einer externen Gedächtnishilfe,*
- *liefert Abstraktionsebenen durch formale Sprachen,*
- *kontrolliert die Auswirkungen von Veränderungen,*
- *hilft, Inkonsistenzen zu vermeiden oder aufzudecken,*
- *erzeugt dynamisches Verhalten aus statischen Beschreibungen,*
- *ermöglicht, die Konsequenzen von Aktionen rückgängig zu machen und fördert damit exploratives und kreatives Verhalten,*
- *bietet verschiedene Perspektiven auf komplexe Informationsstrukturen,*
- *verbirgt durch kontextabhängige und benutzergesteuerte Filter irrelevante Einzelheiten,*
- *lenkt unsere Aufmerksamkeit auf wichtige Informationen oder Ereignisse,*
- *ist ermüdungsfrei wachsam und – führt komplexe und umfangreiche Aktivitäten fehlerfrei durch*

Dies sind nur wenige Faktoren, die bekräftigen, wie ein außerordentliches Potential in der *HCI* verborgen ist und durch entsprechend konzipierte interaktive Systeme zum Tragen kommen kann. Ziel des Designers soll es daher sein, die Fähigkeiten beider Kommunikationspartner optimal auszunutzen und in das System einfließen zu lassen.

2.1.2 Varianten von Systemparadigmen I: Kommunikationspartner

Ein erstes Systemparadigma das Herzog spezifiziert, ist jenes von Mensch und Computer als Kommunikationspartner. Wesentliches Merkmal ist hierbei, dass die interagierende Person mit dem Computer auf eine prinzipielle Art und Weise kommuniziert, die der Auseinandersetzung mit anderen Menschen sehr ähnlich ist. Die Maschine ist aber dennoch in der Rolle des Ausführenden und übernimmt zusätzliche Aufgaben wie Assistenz, Überwachung und Hilfestellung.



Abbildung 2.1: Kommunikationsablauf im Systemparadigma Kommunikationspartner. Aus [24].

Dieser Weg der Interaktion weist bestimmte prinzipielle Grundcharakteristika auf, die signifikant sind. Allem voran ist wichtig, dass die Kommunikationsaktivität in Form eines Dialoges durchgeführt wird, d.h. es steht eine bestimmte Sprache zur Verfügung, die genutzt wird. Dadurch wird ein bestimmtes Rollenverhalten initiiert, wobei der Mensch in erster Linie delegiert und Initiativen ergreift. Diese Kompetenz kann bei fortlaufender Dauer des Dialoges durchaus wechseln.

Die Verwendung von Sprache als Mittel und Weg der Verständigung birgt aber entsprechende Probleme in sich. Die sprachlichen Fähigkeiten des Computers werden sehr oft überschätzt, wodurch es zu Missverständnissen und Mehrdeutigkeiten im Handlungsablauf kommen kann. Ein Sachverhalt, der vor allem durch menschähnliche Darstellungen oder Verwendung der Ich-Form bei Avataren und Assistenten entstehen kann.

Grundsätzlich ist zu sagen, dass das Systemparadigma von Mensch und Computer als Kommunikationspartner schon seit den Anfängen im Interaktionsdesign verfolgt wird. Ansprechende technische und vor allem unmissverständliche Realisierungen erweisen sich allerdings seit jeher als sehr schwierig und sind primär von der richtigen Auswahl der Anwendungsdomäne abhängig.

Als mögliche Umsetzungsvarianten des Systemparadigmas Kommunikationspartner sind zu nennen:

1. Intelligent Agents

Die Grundannahme, dass Computersysteme die Fähigkeit besitzen anspruchsvolle Problemstellungen zu lösen, impliziert eine gewisse Form

von Intelligenz. Aus diesem Grund basiert das System der *Intelligent Agents* auf dem Prinzip des unermüdbaren Helfers, der sowohl Aufgaben entgegennimmt, als auch Unterstützung bei Hilfetemen bietet. Diese intelligenten Benutzerschnittstellen, die durchaus die Fähigkeit der Problemlösung besitzen, sind in wesentlichen Bereichen wohl eher dem Forschungsbereich der künstlichen Intelligenz zuzuordnen. Dennoch ist entscheidend, dass dieses Prinzip sich speziell für jene Menschen hervorragend eignet, die nicht die Fähigkeiten bzw. Möglichkeiten besitzen, sich auf die formalen Kriterien komplexer Benutzerschnittstellen bzw. interaktiver Systeme einzustellen.

2. Avatare und Conversational Agents

Primärer Ansatzpunkt ist bei dieser Variante des Computers als Kommunikationspartner ein gewisses zusätzliches Gefühl der Vertrautheit, das bei der interagierenden Person erzeugt werden soll – vorrangig durch visualisierte, ausdrucksvolle Körperlichkeit. Eine mögliche Umsetzungsvariante zeigt sich z.B. in Form von dreidimensionalen Repräsentationen von Intelligent Agents, die in Lage sind, über Sprache, Gestik, Mimik mit dem Benutzer zu kommunizieren.

Der japanische Wissenschaftler Masahiro Mori beschäftigt sich ebenfalls mit dieser Thematik und weist dabei auf deren Komplexität hin.³ Seiner Ansicht nach werden abstrakte Repräsentationen von *Conversational Agents* von der interagierenden Person wesentlich besser aufgenommen als menschähnliche Darstellungen, die nicht als natürlich empfunden werden.

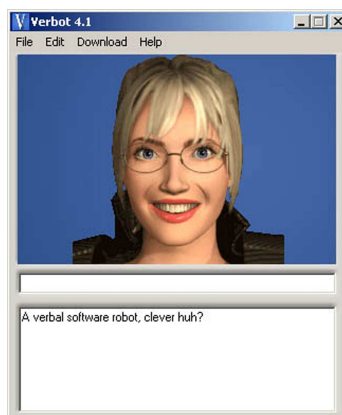


Abbildung 2.2: Beispiel für einen Conversational Agent von Verbot ®. Aus [61].

³Vgl.: Mori Masahiro: *The Buddha in the Robot*. Charles E. Tuttle Publishing, 1982 – In [24]: Herczeg 2006: S. 45

2.1.3 Varianten von Systemparadigmen II: Handlungsräume

Herczeg spezifiziert in seinen Ausführungen ein weiteres Systemparadigma, in dem der Computer verwendet wird, um gemeinsam mit dem Menschen in einen (virtuellen) Handlungsraum einzutauchen. Bedeutend ist hierbei, dass der Fokus weniger auf Austausch und Kommunikation liegt, sondern ein direktes Handeln verlangt wird.

Für diese aktive Teilnahme stehen der interagierenden Person entsprechende Ein- und Ausgabegeräte zur Verfügung, die adäquate Handlungsmöglichkeiten in einem spezifischen Anwendungsraum ermöglichen. Der Computer hat dabei die Rolle der Kreation einer künstlichen und virtuellen Umgebung mit darin befindlichen Objekten, die wahrgenommen und mit entsprechenden Werkzeugen manipuliert werden können.



Abbildung 2.3: Kommunikationsablauf im Systemparadigma Handlungsräume. Aus [24].

Eine mögliche Variante, in der sich ein solches reales Modell manifestiert, ist den Computer als Ressource zu sehen, die vom Menschen als Hilfsmittel verwendet werden kann, um komplexe Problemstellungen zu bearbeiten und zu lösen. Sehr nahe liegt dabei die Betrachtungsweise, den Computer als Werkzeug einzusetzen, das vom Menschen benutzt und bedient wird und die Möglichkeit schafft, zu modellieren und zu bearbeiten. Entscheidend ist, dass ein Arbeitsgegenstand mit einem gewissen Arbeitsziel editiert wird und dabei Arbeitswerkzeuge benutzt werden, um das gewünschte Ergebnis zu erreichen. Besonders wichtig ist die Auswahl des richtigen Instruments, um eine falsche Benutzung zu vermeiden.

Die Kreation und Gestaltung von passenden Tools ist begrifflich sehr stark mit dem Software-Development verbunden. Computerwerkzeuge sind allerdings hochfunktional und damit mit entsprechenden Anforderungen in der Entwicklung verknüpft.

Mögliche Umsetzungsvarianten des Systemparadigmas Handlungsräume sind:

1. Direkt manipulative und metaphorische Systeme

In dieser Form der Mensch-Computer-Kommunikation werden auf Basis einer vorwiegend symbolhaften Sprache, Objekte und Werkzeuge der physischen, realen Welt, durch repräsentative Abbildungen (*Piktogramme*) dargestellt. Entscheidend ist dabei, einen entsprechenden Grad der Abstraktion zu wahren, speziell in der Visualisierung der Be-

dienoberfläche und Interaktionswerkzeuge, um einerseits die technische Komplexität der Darstellung nicht über zu strapazieren und andererseits einen Verlust der metaphorischen Symbolik zu vermeiden.

Zur Bearbeitung bzw. direkten Manipulation wird eine Reihe von Methoden angeboten. Ein wichtiger Aspekt ist in diesem Zusammenhang die Abbildung natürlicher Verhaltensweisen durch Einsatz physikalischer Dialogprinzipien wie *Drag and Drop* oder *Cut and Paste*, um nur zwei zu nennen.

Für eine Charakterisierung dieser Systeme lässt sich auf die Definitionen von Ben Shneiderman verweisen:⁴

- Fortlaufende Darstellung der interessanten Objekte und Aktionen mit bedeutungsvollen visuellen Metaphern
- Physische Aktionen oder Drücken von gekennzeichneten Buttons statt komplexer Syntax (vgl. Kommandosysteme)
- Schnelle, inkrementelle umkehrbare Operationen, deren Effekte auf dem Zielobjekt sofort sichtbar sind.

Diese Grundprinzipien bilden die Basis für diverse Konzepte, die im Rahmen der *HCI* in verschiedenen Anwendungen sehr erfolgreich umgesetzt wurden. Der Erfolg hat dabei unterschiedliche Ursachen. Durch die symbolhafte Implementierung wird auch Gelegenheitsbenutzern die Möglichkeit geboten, die Grundcharakteristika des Systems kennenzulernen und sich schnell zu Recht zu finden. Entscheidend ist allerdings die direkte Abfolge von Aktion und Reaktion. Der Benutzer kann sofort erkennen, ob die von ihm ausgeführte Handlung das gewünschte Resultat gebracht hat, signalisiert durch entsprechendes *Feedback*.

Der vorrangige Einsatzbereich dieser Interaktionstechnik im Bereich von grafischen Benutzeroberflächen bringt aber auch Nachteile und Problemstellungen zum Vorschein. Durch die mehrheitliche Verwendung von typischen *Pointing Devices* zur Interaktion (siehe Abschnitt 2.6.2), erhöht sich die Anzahl der Arbeitsschritte gewöhnlich mit steigender Anzahl der Einzelobjekte, da es normalerweise notwendig ist, Modifikationen von Attributen für jedes Objekt separat durchzuführen. Man stelle sich hierbei z.B. die Visualisierung einer Verzeichnisstruktur auf Basis einer grafischen Benutzeroberfläche vor, in Kombination mit der Aufgabe sämtliche Dateibezeichnungen zu ändern.

2. Virtual, Augmented and Mixed Reality

Die Modellierung und Darstellung von virtuellen Realitäten bietet die Möglichkeit, dem Benutzer eine Vielzahl unterschiedlicher Objekte zur

⁴Vgl. [53]: Shneiderman Ben, Plaisant Catherine: *Designing the User Interface*. Pearson, Addison Wesley, 2005: S. 234

Verfügung zu stellen, die manipuliert werden können. Dadurch wird der Eindruck der Immersion vermittelt, d.h. man hat das Gefühl in eine künstliche Welt einzutauchen. Ein entscheidender Aspekt ist in diesem Zusammenhang, dass oftmals die Prinzipien der Benutzerintegration vom zweidimensionalen Bildschirm in den dreidimensionalen (virtuellen) Raum transformiert werden.

Dadurch ergeben sich zusätzliche Möglichkeiten die Konzeption des Interfaces betrifft, indem man sich die Fähigkeiten des Menschen, die er im Umgang mit physischen Objekten besitzt, zu Nutze macht. Die Benutzer sollen die Möglichkeit bekommen, die Interaktion im System derart zu verinnerlichen, dass nur mehr das Explorieren der künstlichen Welt bzw. das Lösen der gestellten Aufgabe im Vordergrund steht.

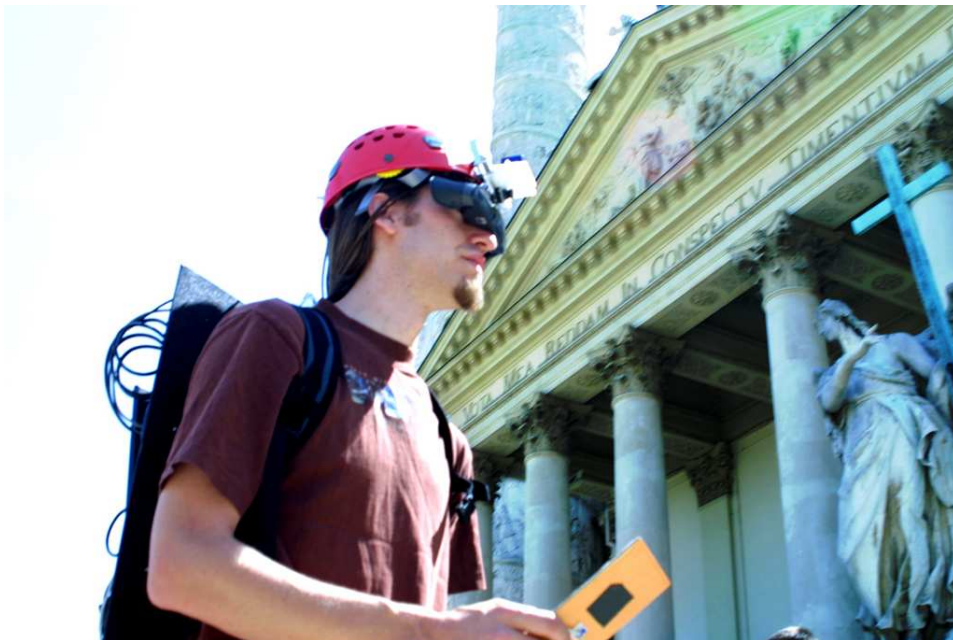


Abbildung 2.4: Beispiel für Augmented Reality. Aus [59].

Fügen sich zur virtuellen Welt in einem ausgewogenen Verhältnis reale Elemente hinzu, so spricht man von *Mixed Realities*. Diese Systeme bieten die Möglichkeit, visuelle, auditive und haptische Wahrnehmungen um entsprechende virtuelle Teilaspekte zu erweitern. Auf Basis einer Stimulierung verschiedener menschlicher Sinne wird versucht, eine erweiterte Immersion zu erreichen. Durch die Integration von realen und virtuellen Elementen wird im Speziellen die Interaktion in diesen Räumen zu einem bedeutenden Erlebnis. Entscheidend ist hierbei, dass ein mentales Modell, welches bereits in der realen Welt gebildet

und angewendet wird, direkt auf das virtuelle Systemkonzept übertragen werden kann.

2.1.4 Varianten von Systemparadigmen III: Mediale Systeme

In den beiden bereits analysierten Systemparadigmen *Kommunikationspartner* und *Handlungsräume* nimmt der Computer eine sehr aktive Rolle ein. Es existieren allerdings auch Anwendungsfälle, in denen die Maschine nicht direkt an der Interaktion beteiligt ist, sondern vielmehr in der Funktion eines Verteilers den Transport von Nachrichten übernimmt. Dadurch wird eine weitere Variante der Interaktion geboten, mit dem Computer in der zentralen Rolle des Vermittlers.

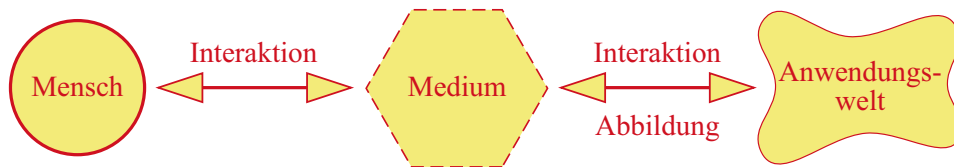


Abbildung 2.5: Kommunikationsablauf im Systemparadigma Mediale Systeme. Aus [24].

Entscheidend ist hierbei, dass der Benutzer Zugriff zu einem Medium bzw. einem Kommunikationspartner erhält, wobei sämtliche Nachrichten entsprechend der unterschiedlichen Anwendungswelten transformiert und übersetzt werden. Diese Form der Interaktion tritt normalerweise in Kombination mit anderen Varianten auf, wie z.B. direkt manipulativen Systemen, was wiederum impliziert, dass Aktion und Reaktion sofort sichtbar sind und durch entsprechende Rückmeldungen signalisiert werden. Das Systemparadigma der medialen Systeme findet primär im Bereich der Hypermediasysteme Anwendung, hierbei insbesondere im World Wide Web.

Bohn und Rohs vom Institut für Informationssysteme der ETH Zürich sprechen in ihrer Publikation zum Thema *Interaktion in Informationssystemen vom Klicken in der realen Welt*⁵ als Interaktionsparadigma. Durch Aktivierung eines Hyperlinks ist es uns möglich, weit entfernte Daten in Form von Dokumenten und Grafiken abzurufen und auf einem lokalen Anzeigegerät darzustellen. Dabei müssen wir uns keine Gedanken über technische Details und weltweite Weitergabe der Informationen machen.

Weiters enthalten Objekte des täglichen Lebens eine Vielzahl von Informationen und können daher in gewisser Weise als physische Hyperlinks betrachtet werden, deren Daten durch die virtuellen Repräsentationen zugänglich

⁵Siehe [7]: Bohn Jürgen, Rohs Michael: *Klicken in der realen Welt – Workshop Mensch-Computer-Interaktion in allgegenwärtigen Informationssystemen*. ETH Zürich, 2001: S. 3

werden. Darüber hinaus bietet sich die Möglichkeit, die realen Objekte in ihrer virtuellen Form mit zusätzlichen Informationen zu versehen, die das eigentliche physische Objekt nicht unbedingt betreffen müssen. Dadurch wird es möglich, Assoziationen zwischen realen Elementen und abstrakten Begriffen zu bilden. Bohn und Rohs führen in diesem Zusammenhang das Beispiel der Wanderschuhe an, die als physische Objekte eine Verbindung zu sämtlichen Informationen über das Wandern ermöglichen können, wodurch sich abstrakte Konzepte bilden lassen. *Durch das „Klicken“ auf die um virtuelle Fähigkeiten und Eigenschaften erweiterten Gegenstände können diese assoziierten Informationen zu einem späteren Zeitpunkt wieder abgerufen werden.*⁶

Mögliche Umsetzungsvarianten des Systemparadigmas Mediale Systeme sind:

1. Information Spaces

Der Computer regelt in seiner Funktion als Vermittler den Zugriff zu Informationsräumen, wobei entscheidend ist, dass Inhalte nicht verändert werden. Den Benutzern werden Möglichkeiten und Varianten zur Verfügung gestellt, die ihnen ein komfortables Navigieren in diesen computergestützten Räumen ermöglichen. Wichtig ist in diesem Zusammenhang, dass die Kreation von Interaktion und Kommunikation vor allem von Art und Weise der Präsentation und Darstellung der Inhalte abhängig ist.

2. Hypermediasysteme

Hypermediasysteme sind grundsätzlich zum Systemparadigma der *Information Spaces* zu zählen. Sie haben die Aufgabe, multimediale Informationen in einem vernetzten Informationsraum zu organisieren.⁷ Hierbei ist aber festzuhalten, dass die Kriterien, um einen Raum der Informationen einem Hypermediasystem zuzuordnen, nicht eindeutig zu definieren sind. Vielmehr sind es wiederum folgende charakteristische Eigenschaften, die ausschlaggebend sind: die textbasierte Datenaufbereitung (*Hypertext*), das vorrangige Interagieren mittels zeitorientierten Werkzeugen (vgl. Pointing Devices) und vor allem das Navigieren in diesem Datenraum durch passende Verweise im Text auf verwandte Themen (Englisch: *Links*). Dadurch sind die Benutzer in der Lage, Assoziationen zwischen den einzelnen Informationselementen herzustellen. Als Beispiel für ein solches System ist das *Internet* zu nennen.

In größeren Hypermediasystemen besteht die Gefahr, sehr schnell die Orientierung zu verlieren. Dies kann verhindert werden, indem zusätzliche Übersichtsinformationen zur Verfügung gestellt werden, wodurch

⁶Siehe [7]: Bohn, Rohs, 2001:S. 3

⁷Vgl.: Bogaschewsky R.: *Hypertext-/Hypermedia-Systeme: Ein Überblick*. (In: Informatik Spektrum, 15(3), 1992: S. 127-143) – In: [24]: Herczeg 2006: S. 79

der Benutzer, ausgehend von seinem eigenen Standort, in der Lage ist, eine Exploration des Informationsraumes zu planen und zu steuern.

3. Communication Systems

In medialen Systemen nimmt der Computer die Rolle eines Vermittlers von Kommunikation zwischen Mensch und Maschine ein, d.h. er reduziert seine Rolle als Handlungs- bzw. Kommunikationspartner und übernimmt die Aufgabe des Transportmediums. Diese Form der Interaktion findet man vorrangig umgesetzt in Systemen des *Instant Messaging*, wie z.B. *ICQ* oder *MSN Messenger*. Entscheidend ist, dass die Nachrichten zwischen den Informationswelten zwar transformiert aber inhaltlich nicht verändert werden.

2.1.5 Zusammenfassung

Die Konzepte, Theorien, Definitionen und nicht zuletzt die persönlichen Ansichten, die Herzeg im Zusammenhang mit Interaktionsdesign formuliert, sind sehr stark vom Modell des Computers als Kommunikationspartner geprägt. Er geht davon aus, dass Mensch und Maschine in der Lage sind, ähnlich wie Menschen miteinander zu kommunizieren. Der entstehende Dialog ist dabei an einen gewissen Handlungsraum gebunden, in dem der Benutzer seine Aktionen ausführt. Im Rahmen der Konzeptionalisierung und Gestaltung von Kommunikations- und Handlungsräumen spricht Herzeg von so genannter Mediengestaltung, in besonderer Hinsicht auf den ubiquitären Einsatz von multimedialen Computersystemen.

Diese Sachverhalte bilden dabei die Grundlage für die unterschiedlichen Basismodelle der Interaktion, die präsentiert werden. In diesem Zusammenhang spezifiziert Herzeg den Begriff des Systemparadigmas, das im Grunde genommen in seinen unterschiedlichen Ausprägungen die Varianten und Systemkonzepte der Mensch-Computer-Kommunikation beschreibt. Diese Paradigmen stellen zum einen die Grundlage zur Analyse bestehender interaktiver Systeme dar und bilden zum anderen die Basis für neue modifizierte Konzepte. Entscheidend ist für Herzeg aber, dass diese Theorien das Fundament repräsentieren, wie Computersysteme von ihren Benutzern, bezogen auf Anwenderfreundlichkeit, wahrgenommen werden.

Die in dieser Arbeit präsentierten Systemparadigmen lassen sich strukturieren in folgende Bereiche:

- **Kommunikationspartner**
 - Intelligent Agents
 - Avatare und Conversational Agents

- **Handlungsräume**

- Direkt manipulative Systeme und metaphorische Systeme
- Virtual, Augmented and Mixed Realities

- **Mediale Systeme**

- Information Spaces
- Hypermediasysteme
- Communication Systems

Unterscheidungen sind dabei nicht immer eindeutig zu treffen, da ähnliche Grundcharakteristika und oftmals auch gewisse Abhängigkeiten existieren. Entscheidend ist vielmehr, dass durch den adäquaten Einsatz dieser Konzepte Computeranwendungen benutzerfreundlich, intuitiv und transparent gestaltet werden können. Im Rahmen des Interaktionsdesigns ist es daher von essentieller Bedeutung, die richtigen Systemparadigmen auszuwählen, weiters gestalterische und technische Konkretisierungen auszuprägen und – vielleicht auch durch Kombination bestehender Konzepte – neue Varianten ansprechender und interessanter Interaktion zu schaffen. Herzeg spricht in diesem Zusammenhang von *Gestaltungsmustern*⁸ (in früheren Publikationen seinerseits auch als *Dialogparadigmen*⁹ bezeichnet), die etablierte Lösungsmöglichkeiten zur Umsetzung der bereits besprochenen Systemkonzepte darstellen. Hierbei ist vor allem auch die Dynamik des Interaktionsdesigns zu erkennen, denn etablierte Konzepte werden zwar einerseits von neuen Ideen in der *HCI* verdrängt, bilden aber dabei die Grundlage für diese. Wie in der Einleitung zu diesem Kapitel bereits erwähnt, wird der Begriff des Interaktionsparadigmas in der Literatur zur Definition verschiedener Sachverhalte herangezogen. Prägt Herzeg mit dem Konzept des Systemparadigmas einen Ansatz, der sehr stark auf kommunikationswissenschaftlichen Elementen beruht, so führt er auch konkrete Beispiele an.

Die von Herzeg präsentierten Konzepte und Modelle sind ein wesentlicher Beitrag zu dieser Arbeit, um der Definition des Begriffes Interaktionsparadigma näher zu kommen. Dies liegt in erster Linie an seiner klar strukturierten Vorgehensweise in der Präsentation der Thematik und an der zusätzlichen Illustration mit passenden Beispielen.

⁸Siehe [24]: Herzeg, 2006: S. 71

⁹Siehe [23]: Herzeg Michael: *Software-Ergonomie*. Oldenbourg Wissenschaftsverlag GmbH, 2005: S. 14

2.2 Theorien, Modelle und Interaktionsstile nach Shneiderman

It's time to get angry about the quality of user interface design
(Ben Shneiderman)

Ben Shneiderman ist zum aktuellen Zeitpunkt Professor am Institut für *Computer Science* und Mitglied der Institute für *Advanced Computer Studies* und *System Research* an der Universität Maryland. Zusätzlich ist er ein Gründungsmitglied des *Human-Computer Interaction Laboratory*. Seine zahlreichen Bücher und Publikationen machen ihn zu einem international anerkannten Wissenschaftler auf dem Gebiet der *HCI*. Er hat im Rahmen seiner Tätigkeit zahlreiche Grundprinzipien und Leitlinien des Interaktionsdesigns definiert, erläutert oder mitgestaltet. Entscheidend ist hierbei, dass er sich seit Beginn der kommerziellen Computernutzung intensiv mit der Thematik auseinandersetzt und dadurch über ein sehr breites Fachwissen und große Erfahrung verfügt. Seinen Theorien und Konzepten liegt zudem eine lange Forschungs- und Entwicklungszeit zu Grunde. Ben Shneiderman zählt ohne Zweifel zu den bedeutenden Personen auf dem Gebiet der *HCI*. In den Publikationen, die für diese Arbeit herangezogen wurden (Literaturverzeichnis: [49] – [54]), beschäftigt er sich in erster Linie mit dem Prozess des Interaktionsdesigns im Rahmen der Softwareentwicklung und erläutert dabei interessante Ansätze, Theorien und Richtlinien. Die dabei von ihm spezifizierten Konzepte und Grundregeln setzen vorrangig in den Bereichen des User-Interface-Design und der systemgestützten Benutzerführung an. Entscheidend für die Thematik der Interaktionsparadigmen ist aber seine Auseinandersetzung mit den Methoden und Varianten der Benutzerintegration. Shneiderman spricht in diesem Zusammenhang von verschiedenen primären Interaktionsstilen, die dem Systemdesigner zur Verfügung stehen. Anders als Herczeg, der ja vorrangig auf die verschiedenen Ausprägungen von Mensch-Computer-Systemen eingeht, sind Shneiderman's Theorien sehr stark von Aspekten der *Software-Usability* geprägt. Er versucht dabei einen Überblick zu liefern, auf welche Art und Weise der Benutzer in den Aktionsraum eines interaktiven Systems eingebunden wird und verzichtet auf zusätzliche systembezogene Erläuterungen.

2.2.1 Interaktionstheorien: Object-Action Interface Model

Die Konzepte und Methoden, die Shneiderman im Rahmen seiner Auseinandersetzung mit Interaktionsdesign anführt, sind sehr stark von der Arbeit mit Modellen und Theorien geprägt. Er definiert in diesem Zusammenhang den Begriff der so genannten *High-Level-Theorien*.¹⁰ Grundsätzlich sollen

¹⁰Siehe [51]: Shneiderman Ben: *User Interface Design*. mitp-Verlag, 2002: S. 74

diese unterstützend wirken, um eine Beschreibung der vielfältigen Eigenschaften von interaktiven Systemen zu ermöglichen.

Da sich Interaktionsdesign im Spannungsfeld verschiedener wissenschaftlicher Teilgebiete befindet, existieren sehr viele Konzepte, die mehr oder weniger wetteifern, um von unterschiedlichen Designern unterstützt und angewendet zu werden. Dies ist zum einen sehr positiv für die Disziplin der *HCI*, da ständig neue Ansätze entstehen oder bereits vorhandene modifiziert und weiterentwickelt werden. Zum anderen werden dadurch sehr intensive Anforderungen an die Designer gestellt, die mit den andauernden Veränderungen Schritt halten müssen. Entscheidend aber ist in diesem Zusammenhang, dass Theorien und Modelle für Shneiderman die Grundlage darstellen, um komplexe Zusammenhänge in der Kooperation zwischen Mensch und Computer zu beschreiben und zu deuten.

Ähnlich wie Herzog geht Shneiderman in seinen Ausführungen in [49] auf eine Aufgabenteilung zwischen Mensch und Computer ein, wodurch die jeweiligen Fähigkeiten zum Tragen kommen sollen. Diese Thematik wird allerdings in [51] nur mehr am Rande behandelt und mehrheitlich durch Erläuterungen des Zusammenhanges von Objekt und Aktion, dem so genannten *Object-Action Interface Model*¹¹, verdrängt.

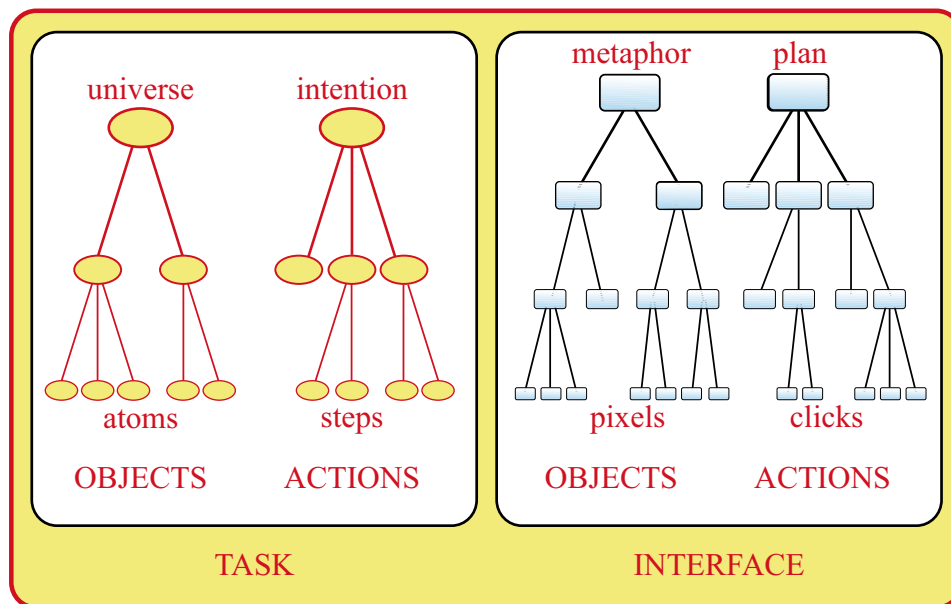


Abbildung 2.6: *Object-Action Interface Model* – Objekte und Aktionen sind organisiert in einer hierarchischen Struktur, wodurch die Abhängigkeiten zwischen den Elementen sichtbar werden. Aus [53].

¹¹Siehe [53]: Shneiderman, 2005: S. 96

Ausschlaggebend ist hierbei, dass auf Basis der direkten Manipulation, modifizierbare Objekte durch entsprechende Darstellungen repräsentiert werden und zusätzlich Werkzeuge zur Verfügung stehen, die es dem Benutzer erlauben, bestimmte Aktion durchzuführen. Die Objekte sind dabei als Informationsbehälter zu sehen, die in weiterer Vertiefung in kleinere Einheiten zerlegt werden können. Besonders wichtig ist in dieser Arbeitsweise ein Verständnis der Aufgabe in Kombination mit einer klaren Vorgehensweise. Schlussendlich liegt es am Designer die Zusammenhänge zwischen Objekt, Aktion und Schnittstelle in adäquater Art und Weise umzusetzen. Im Folgenden werden hierzu Beispiele für Varianten der *HCI* präsentiert, die von Shneiderman als so genannte *Interaktionsstile* bezeichnet werden. Grundsätzlich sind gewisse Parallelen zu den Konzepten und Beispiele von Michael Herzeg gegeben. Aus diesem Grund werden nachfolgend nur noch ergänzende Elemente präsentiert.

2.2.2 Varianten von Interaktionsstilen I: Direkt manipulative und virtuelle Systeme

Ziel des Interaktionsdesign sollte es u.a. sein, ein Gefühl der Zufriedenheit beim Benutzer hervorzurufen. Positive Reaktionen zeigen sich dabei durch verschiedene Fähigkeiten, die der Anwender in der Auseinandersetzung mit dem System erlangen kann, wie z.B.:¹²

- *Leichtigkeit beim Erlernen des Systems*
- *Kompetenz bei der Durchführung der Aufgaben*
- *Vertrauen in die eigenen Fähigkeiten*
- *Beherrschung des Interfaces*
- *Erforschung des Systems*
- *Freude bei der Verwendung des Systems*

Diese Gefühle sind u.a. Faktoren, die über den Erfolg eines Systems und die Akzeptanz beim Benutzer entscheiden. Nach Shneiderman sind die elementaren Ideen von Benutzersystemen, die solche Ansätze transportieren, stark an die Sichtbarkeit der bedeutenden Objekte, Werkzeuge und Aktion gebunden. Die Bedienung des Systems wird intuitiver und langen Phasen des Erlernens der korrekten Bedienung wird vorgebeugt. Entscheidend ist in diesem Zusammenhang, dass sich die direkte Manipulation von Objekten und das damit verbundene unmittelbare Erkennen des Zusammenhangs von Aktion und Reaktion, als brauchbares Modell bewährt hat. In diesem Kontext ist wiederum auf das *Object-Action Interface Model* von Shneiderman zu verweisen, das die grundlegenden Eigenschaften der direkten Manipulation illustriert. Shneiderman definiert hierzu mehrere Grundprinzipien, die

¹²Siehe [51]: Shneiderman, 2002: S. 227

ein eindeutiges Bild der direkten Manipulation konstruieren. Entscheidende Erkenntnis aus diesem Sachverhalt ist, dass die Visualisierung von Information und Interaktion ein wesentliches Konzept darstellt, um den Benutzern in möglichst kurzer Zeit gewisse Grundkompetenzen der Handhabung zu vermitteln.

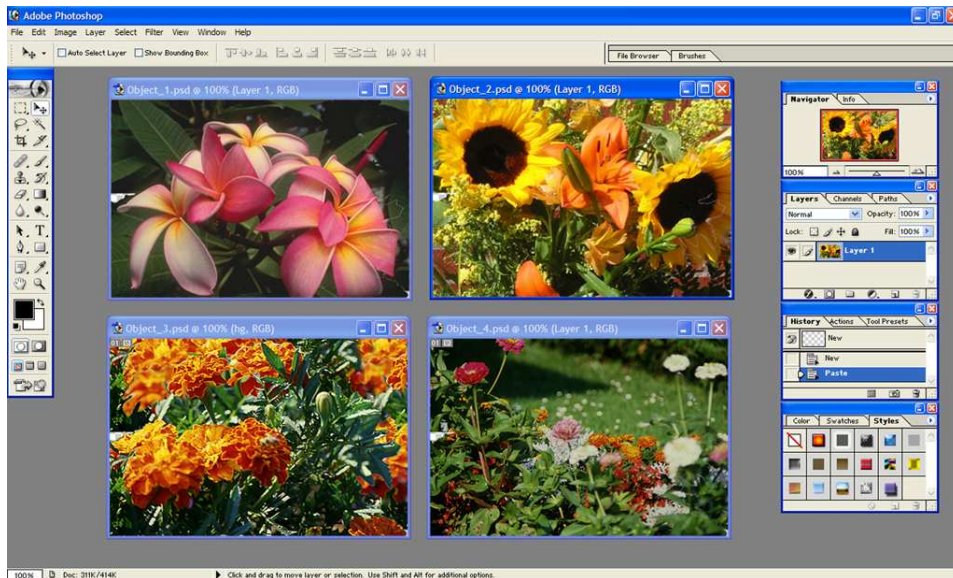


Abbildung 2.7: Ein direkt manipulatives System am Beispiel von Adobe Photoshop® – Sämtliche Interaktionskomponenten werden ausschließlich durch grafische Repräsentationen dargestellt. Einzelne Objekte, in diesem Fall Abbildungen von Blumen, können mit Hilfe von Werkzeugen modifiziert werden. Aktion und Reaktion sind dabei unmittelbar zu erkennen.

Die direkte Manipulation ist aber auch mit gewissen Problemen behaftet:

- Durch das Arbeiten mit grafischen Benutzeroberflächen werden entsprechende Anforderungen an das User-Interface-Design gestellt. Hierbei ist eine ständige Verfügbarkeit der primären Interaktionswerkzeuge der gewünschte Zustand.
- Weiters ist das Arbeiten mit grafischen Repräsentationen mit Schwierigkeiten behaftet. Die visuelle Darstellung ist zwar sehr schnell zu deuten, kann jedoch irreführend sein.
- Das vorrangige Interagieren mittels Zeigewerkzeug kann für den geübten Benutzer einige Nachteile mit sich bringen, da in manchen Situationen das Eintippen eines Kommandos der effektivere Weg der direkten Interaktion ist.

Wie sich hier zeigt, ist der Erfolg eines interaktiven Systems nicht allein durch die Verwendung der direkten Manipulation als Interaktionsstil gewährleistet. Vielmehr ist ein ausgezeichnetes Design in Kombination mit einer adäquaten Funktionalität ein ausschlaggebendes Argument, was die Akzeptanz des Systems betrifft.

2.2.3 Varianten von Interaktionsstilen II: Menüauswahl und Formularfelder

Menüsysteme und die in weiterer Folge beschriebenen Formulareingabefelder, können aber unter gewissen Umständen interessante Alternativen zu den direkt manipulativen Systemen darstellen. Die Navigation auf der Bedienoberfläche wird bei dieser Interaktionsvariante durch eine mehr oder minder umfangreiche Menüstruktur ermöglicht.

Die einzelnen Menüebenen sind dabei in einer strengen Hierarchie organisiert, d.h. durch entsprechende Verzweigungen (Vernetzungen) zwischen den Elementen gelangt man zur gewünschten Eingabemaske.

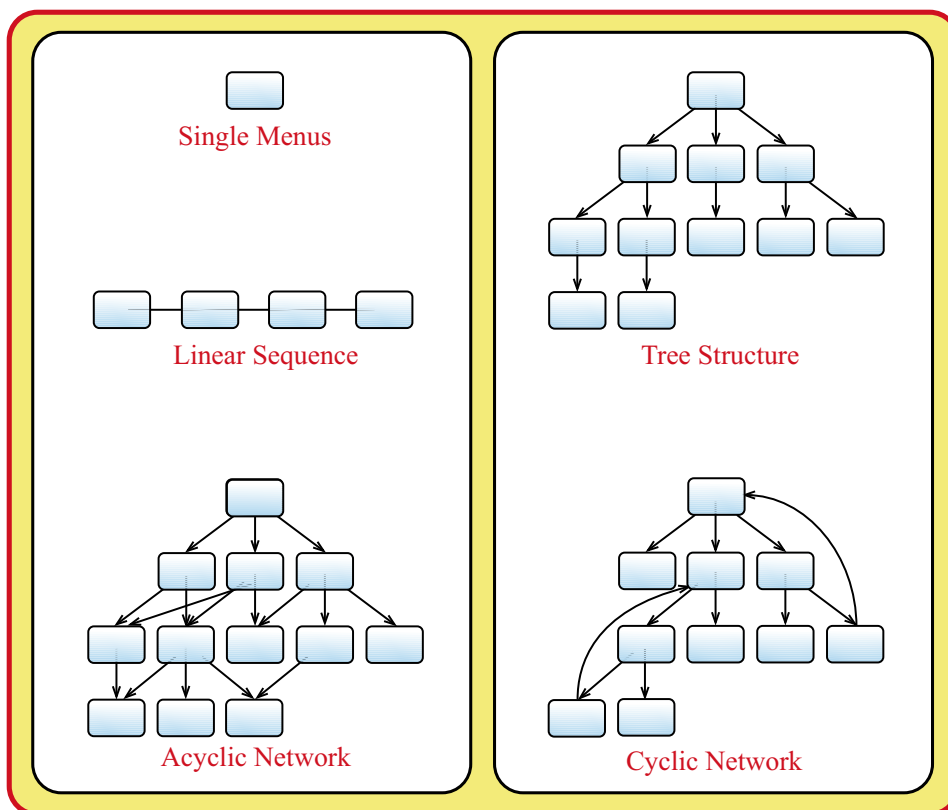


Abbildung 2.8: Varianten von Menüsystemen – von einfachen Single Menus bis hin zu komplexen zyklischen und azyklischen Strukturen. Aus [53].

Auf der Hand liegt, dass die Komplexität mit der Ebenentiefe bzw. der Intensität der Vernetzung stark zunimmt, was zu einer Desorientierung des Benutzers führen kann. Aus diesem Grund ist es notwendig, ein gewisses Maß an Unterstützung zu bieten. Zum einen durch eine klar definierte Darstellung des Interaktionsweges, wodurch eine eindeutige Rückverfolgung der Verbindungen durch die Menühierarchie möglich wird und dadurch die Orientierung stützt. Zum anderen wird durch entsprechende Ausfüllhilfen (z.B. durch erklärende Angaben zu den Formularinhalten, sequentielle Weiterführung des Cursors entsprechend der vom System vorgegebenen Ausfüllreihenfolge) dem Benutzer eine zusätzliche Hilfestellung angeboten.

Im Zusammenhang mit dem *Object-Action Interface Model* liegt der Schlüssel zum Design einer praktikablen Menüstruktur wiederum in der Unterteilung in Objekte und Aktionen, die auf diese angewendet werden können. Die bedeutenden Elemente der Benutzeroberfläche werden dabei in kategorisierten Hauptgruppen mit entsprechenden Untergruppen organisiert. Die Darstellung und Anwendung der vorhandenen Werkzeuge und Schnittstellenobjekte kann in beliebiger Form erfolgen. Entscheidend ist, dass eine hierarchische und klare Organisation der Benutzeroberfläche gefunden wird, die dem Anwender durch Wiedererkennungsmomente ein leichteres Navigieren in der Menüstruktur ermöglicht.

In der Praxis bewähren sich Menüsysteme sehr gut, speziell wenn es um die Interaktion in komplexen Informationssystemen geht. Für gewisse Formen der Dateneingabe (persönliche Angaben z.B. auf Websites) sind Formularfelder der adäquate Interaktionsstil.

Ein interessanter Ansatz in Bezug auf den Dialog zwischen Mensch und Computer ist hierbei, dass aus Sicht des Benutzers ein gewisses Kontrollmoment gegeben ist, da ausschließlich der Anwender über Art und Weise der Dateneingabe entscheidet. In [51] werden dezidierte Designrichtlinien zur Gestaltung von Formulareingabemasken angeführt, die hier auszugsweise präsentiert werden.

Design-Richtlinien für Formulareingaben nach Shneiderman:¹³

- *Aussagekräftige Titel*
- *Verständliche Instruktionen*
- *Logische Gruppierung der Felder*
- *Verstraute Feldbezeichnungen*
- *Sichtbarer Platz für Dateneingabefelder*
- *Fehlerkorrektur für einzelne Zeichen und ganze Felder*
- *Fehlermeldungen für inakzeptable Werteingaben*
- *Erklärende Meldungen für die Dateneingabe*

¹³Siehe [51]: Shneiderman, 2002: S. 312

Durch entsprechendes Design der Benutzeroberfläche lassen sich auf Basis von Menüauswahl und Formulareingabe Applikationen schaffen, die intuitiv und anwenderfreundlich konzipiert sind. Hier liegt aber auch die Problematik dieser Systeme verborgen. Für verschiedene Benutzergruppen werden zwar ähnliche Gestaltungsansätze umgesetzt, allerdings sind die Bedürfnisse je nach Anwender wiederum unterschiedlich. Aus diesem Grund sollte in einem Menü- bzw. Formularsystem die Möglichkeit geboten werden, die Struktur des Interfaces auf persönliche Ansprüche hin anzupassen.

Interessante Aspekte zur Designproblematik werden in [54] genauer angesprochen, wobei hierbei der Fokus primär auf der Mensch-Computer-Kommunikation in datenbankbasierten Systemen liegt. Speziell in Anwendungen des WWW sind Formularsysteme eine beliebte Variante der Interaktion, allerdings wird hierbei die mehrheitlich unübersichtliche Art und Weise der Datenvisualisierung bemängelt. Formularbasierte Datenbankabfragen liefern in der Regel entweder unbefriedigende bis gar keine Treffer bzw. eine Vielzahl von Ergebnissen, was wiederum die nötige Genauigkeit vermissen lässt. In diesem Zusammenhang spricht Shneiderman von einer mangelnden visuellen Repräsentation der Datenwege in Datenbanken, d.h. es werden keine Indikationen geboten, die über das zu erwartende Ergebnis vorab informieren. Das System entmachtet zudem quasi den Benutzer für gewisse Zeit(Such)-perioden und bietet darüber hinaus oftmals keine Leitlinie, um die Suche zu modifizieren, falls das gewünschte Ergebnis nicht erzielt wird. Hierbei kann allerdings Abhilfe geschaffen werden durch so genannte *Query Previews*¹⁴, die zum einen Information über die Verteilung bzw. Streuung spezieller, selektierter Attribute des Systems bieten und zum anderen kontinuierliches Feedback liefern, während die Abfrage durchgeführt wird.

Zusammenfassend bleibt zu sagen, dass Menüauswahl und Formularfelder schon seit Beginn der kommerziellen Computernutzung Anwendung finden, aber von neueren, moderneren Konzepten verdrängt wurden. Nicht zuletzt erleben sie aber speziell durch ihren Einsatz im Bereich des WWW eine Renaissance.

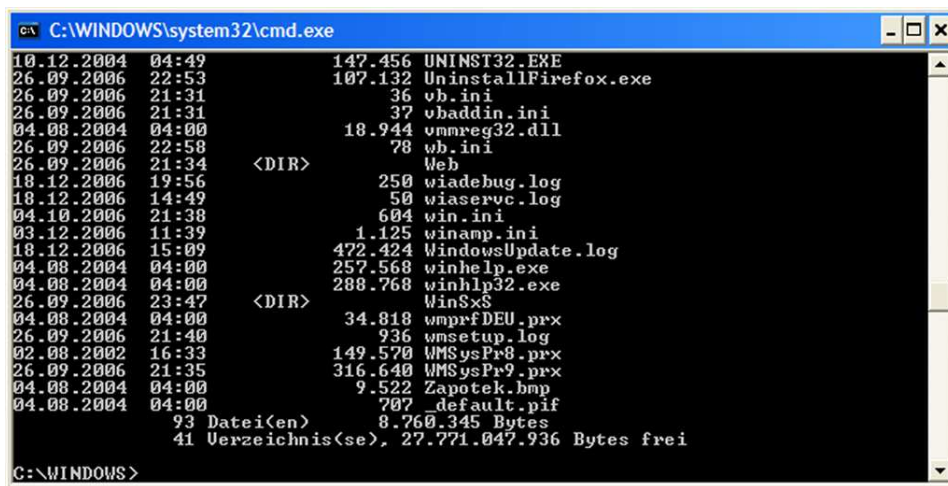
2.2.4 Varianten von Interaktionsstilen III: Befehlssprachen und Natürliche Sprachen

In der chronologischen Reihenfolge der Sprachentwicklung gab es immer wieder technische Innovationen, die einen bemerkenswerten Ansporn in Hinblick auf den Fortschritt verursachten. Eine solche Position nimmt auch der Computer ein, dessen Einsatz zur Entwicklung und zum Design neuer formaler Sprachen geführt hat. Diese Sprachen müssen dabei die angegebenen Aufgaben der Anwender abbilden und ein gewisses Maß an Kommunikation bieten

¹⁴Siehe [54]: Tanin Egemen, Lotem Amnon, Haddadin Ihab, Shneiderman Ben, Plaisant Catherine, Slaughter Laura: *Facilitating Data Exploration with Query Previews – A Study of User Performance and Preference*. University of Maryland, 1999: S. 2

und sollen aber gleichzeitig innerhalb einer Softwareapplikation ordnungsgemäß verarbeitet werden können.

Shneiderman trifft hierzu eine Unterscheidung zwischen Computersprachen, die in einer interaktiven bzw. in einer nicht-interaktiven Umgebung eingesetzt werden. Er verwendet in diesem Kontext den Begriff der Befehlssprachen, die in unterschiedlichen Ausprägungen zur Interaktion mit Systemen verschiedenster Art angewendet werden. Ein signifikantes Kriterium ist hierbei, dass Befehle eine bestimmte Struktur besitzen, was unterschiedliche Variationen von Befehlsketten ermöglicht. Zudem wird die Korrektheit von Eingaben mit Feedback in entsprechender Form signalisiert. Eine bedeutungsvolle Struktur ist dabei auch der Schlüssel, um ein ansprechendes und praktikables Bedienkonzept zu realisieren.



```

C:\WINDOWS\system32\cmd.exe
10.12.2004 04:49      147.456 UNINST32.EXE
26.09.2006 22:53      107.132 UninstallFirefox.exe
26.09.2006 21:31           36 vb.ini
26.09.2006 21:31           37 vbaddin.ini
04.08.2004 04:00      18.944 vmmreg32.dll
26.09.2006 22:58           78 wb.ini
26.09.2006 21:34      <DIR>      Web
18.12.2006 19:56       250 wiadebug.log
18.12.2006 14:49          50 wiaservc.log
04.10.2006 21:38       604 win.ini
03.12.2006 11:39         1.125 winamp.ini
18.12.2006 15:09     472.424 WindowsUpdate.log
04.08.2004 04:00     257.568 winhlp.exe
04.08.2004 04:00     288.768 winhlp32.exe
26.09.2006 23:47      <DIR>      WinSxS
04.08.2004 04:00     34.818 wmpwfDEU.prx
26.09.2006 21:40          936 wmssetup.log
02.08.2002 16:33     149.570 WMSysPr8.prx
26.09.2006 21:35     316.640 WMSysPr9.prx
04.08.2004 04:00          9.522 Zapotek.bmp
04.08.2004 04:00          707 _default.pif
          93 Datei(en)
          8.760.345 Bytes
          41 Verzeichnis(se), 27.771.047.936 Bytes frei
C:\WINDOWS>

```

Abbildung 2.9: Interaktion mittels Befehlssprachen – Command Prompt bei Microsoft Windows XP (®).

Die Interaktion mittels Befehlssprachen ist für Gelegenheitsnutzer unpassend. Selbst für routinierte Anwender ist die syntaktisch richtige Eingabe von Befehlen eine fehleranfällige Tätigkeit. Als Vorteile sind wiederum Geschwindigkeit und Flexibilität, sowie die Unabhängigkeit von speziellen Eingabegeräten zu nennen.

Ziel in der fortwährenden Entwicklung der Befehlssprachen ist laut Shneiderman eine Interaktion mit natürlicher Sprache, die sowohl die menschlichen als auch die maschinellen Anforderungen erfüllt. Die Idealvorstellung dreht sich dabei um Maschinen, die in Lage sind, natürliche, sprachliche Laute zu verarbeiten. Shneiderman verwendet in diesem Zusammenhang den Begriff *Natural Language Interaction (NLI)*.¹⁵ Die großen Probleme der *NLI* werden

¹⁵Siehe [51]: Shneiderman, 2002: S. 346

offensichtlich, wenn man einen Vergleich mit dem *Object-Action Interface Model* vornimmt. Es ist sehr schwierig, klare Informationen über die Strukturierung von Daten zu vermitteln (z.B. im Vergleich zu Menüsystemen, die durch die hierarchische Abbildung einen klaren Weg der Interaktion vorgeben). Zudem ist eine gewisse visuelle Repräsentation unabdingbar, um gewisse Grundkonzepte der Interaktion unmissverständlich darzustellen. Eine optimale Umsetzung dieser Interaktionsvariante wäre laut Shneiderman daher in der Kombination mit bereits bestehenden Interaktionsstilen gegeben. Das Ziel wäre eine klare strukturierte Gestaltung der Kommunikationsumgebung zusammen mit natürlicher Sprache als Eingabemedium. Zum momentanen Zeitpunkt ist diese Situation bei weitem noch eine Wunschvorstellung, da sämtliche Spracheingabesysteme beträchtlichen Aufwand beim Bearbeiten von Input und Output verursachen. Weiters stellt vor allem auch die Abbildung von sprachlichen Sonderfällen, sowie emotionalen und kulturellen Kontexten, eine ungelöste Aufgabe dar. Grundsätzlich bleibt aber zu sagen, dass *NLI* durchaus viel versprechende Möglichkeiten bietet, die es in Form von entsprechenden Designkonzepten umzusetzen gilt.

2.2.5 Zusammenfassung

Ben Shneiderman hat im Rahmen seiner Auseinandersetzung mit dem Themengebiet der *HCI* maßgebliche Akzente gesetzt. Speziell durch seine Arbeit mit Modellen und Theorien schafft er die Möglichkeit, die Abstraktheit der Thematik auf ein Minimum zu reduzieren, um damit verbunden klare und eindeutige Richtlinien des Interaktionsdesigns zu spezifizieren. Nicht umsonst wird in der Literatur mehrheitlich auf seine Konzepte hingewiesen, die für viele Designer und Wissenschaftler die Grundlage zu Kreation ansprechender Benutzerinteraktion darstellen.

Hierbei ist es allerdings sehr wichtig klar zu stellen, dass Shneiderman sich vorrangig mit Problemstellungen im Bereich der Softwareergonomie auseinandersetzt. Er geht dabei weniger auf Aspekte der Systementwicklung ein, sondern befasst sich vielmehr mit den Varianten des User-Interface-Designs und der systemgestützten Benutzerführung. Er hat im Rahmen seiner Tätigkeit Grundprinzipien und Leitlinien des Interaktionsdesign definiert, erläutert oder mitgestaltet und in Form von zahlreicher Publikationen veröffentlicht, von denen einige ausgewählte auch für diese Arbeit herangezogen wurden (Literaturverzeichnis: [49] – [54]).

Seine Modelle und Konzepte basieren dabei auf bestimmten Grundansätzen. Er definiert Theorien und Prinzipien, um die komplexen Zusammenhänge im Verhältnis von Mensch und Computer zu deuten und greifbar zu machen. Darüber hinaus präsentiert er verschiedene Formen der Benutzerinteraktion, die speziell für die Thematik der Interaktionsparadigmen äußerst interessant sind. Er spricht in diesem Zusammenhang von so genannten Interaktionsstilen, die im Grunde genommen die primären und gängigen Basismodelle

der *HCI*, wohlgermerkt auf Software-Ebene, repräsentieren. Dadurch wird die Möglichkeit geboten, interaktive Systeme ordnungsgemäß zu evaluieren und zu klassifizieren.

Innerhalb dieser Arbeit wurden folgende Interaktionsstile nach Shneiderman präsentiert und analysiert:

- Direkte Manipulation
- Menüauswahl
- Eingabefelder
- Befehlssprachen
- Natürliche Sprachen

Im Rahmen des Interaktionsdesigns ist es von essentieller Bedeutung, die passende Interaktionsform bzw. den passenden Interaktionsstil auszuwählen. Entscheidend ist aber, dass die hier angeführten Modelle und Konzepte auch nur Ausschnitte darstellen von einer Vielzahl von Optionen, die dem Designer zur Verfügung stehen. Speziell durch die multiplen Möglichkeiten des Einsatzes von Computern und die Entwicklung im Bereich alternativer Input-Devices, sind die präsentierten Formen nur mehr als grundlegende Konzepte zu betrachten, die in weiter entwickelten, modifizierten Formen Anwendung finden.

2.3 Interaktionsparadigmen und Interfacemetaphern nach Jenny Preece

By Interaction design we mean:

designing interactive products to support people in their everyday lives (Jenny Preece)

And

Creating user experience that enhance and extend the way people work, communicate and interact (Yvonne Rogers)

Jenny Preece ist Professorin für Informationssysteme an der Universität Maryland, wo sie auch für fünfeinhalb Jahre bis Herbst 2002 die Vorsitzende des *Departments for Information Systems* war. Bevor sie 1996 der Universität von Maryland beitrug, war sie u.a. Professorin an der British Open University und Mitglied des Research-Teams für Human-Computer-Interaction am *HCI-Research-Center* der South Bank University in London. Auf dem Gebiet des Interaktionsdesigns wurde sie bekannt durch Bücher und Publikationen zur Thematik, sowie durch ihre Forschungsarbeit im Bereich

Online-Communities und den damit verbundenen Zusammenhängen zwischen Aspekten der Benutzfreundlichkeit und sozial-gesellschaftlichen Anhaltspunkten. Für die Erstellung ihrer Bücher und Publikationen, auf die inhaltlich innerhalb dieser Arbeit verwiesen wird, arbeitet sie vorrangig mit ihren Kolleginnen Yvonne Rogers und Helen Sharp zusammen (Literaturverzeichnis: [32], [42], [43], [53], [54]). Das weit reichende Wissen aller drei Autorinnen auf dem Gebiet der *HCI*, in Kombination mit ihren Fähigkeiten in anderen Disziplinen, wie Software-Engineering und kognitiven Wissenschaften, befähigt sie, Ideen, Konzepte und Theorien zu formulieren, die das breite Spannungsfeld des Interaktionsdesigns abdecken.

Die Thesen von Jenny Preece werden in dieser Arbeit vor allem deshalb angeführt, weil sie eine interessante Alternative zu den bisher präsentierten Definitionsvarianten des Begriffes Interaktionsparadigma darstellen. Entscheidend ist dabei, dass sie die Thematik als Teil der konstruktiv, konzeptionellen Auseinandersetzung mit Interaktionsdesign sieht und dabei vorrangig versucht, das Interaktionsparadigma als Form der Kommunikation zwischen Mensch und Computer, sowohl aus Sicht der Benutzer als auch der Designer, zu betrachten.

2.3.1 Grundzüge von konzeptionellen Modellen

Die Arbeit von Preece im Bereich des Interaktionsdesigns ist mehrheitlich geprägt durch die Auseinandersetzung mit Metaphern, Paradigmen und konzeptionellen Modellen. Dadurch lässt sich eine genaue Beschreibung eines interaktiven Systems, in Bezug auf integrierte Ideen und Konzepte, formulieren. Es wird spezifiziert, welche Möglichkeiten das System bietet, wie es aussieht, wie es sich verhält, auf welche Art und Weise sich der Benutzer damit auseinandersetzen kann. Um solche Modelle erstellen zu können, bedarf es einer sehr umfangreichen und genauen Untersuchungsarbeit, um einerseits die Bedürfnisse der Benutzer erkennen zu können und andererseits ein passendes Design zu kreieren, das sowohl in punkto Funktionalität als auch Authentizität die Anforderungen erfüllt und Maßstäbe setzt.

Preece definiert hierzu eine Reihe von konzeptionellen Modellen, die in Verbindung mit bestimmten, von Benutzern bevorzugten Aktivitäten auftreten. Dies sind:¹⁶

- *Instructing* – Anweisungen geben
- *Conversing* – Konversation führen
- *Manipulating and Navigating* – Manipulieren und Navigieren
- *Exploring and Browsing* – Erkunden und Durchsuchen

¹⁶Vgl. [42]: Preece Jenny, Rogers Yvonne, Sharp, Helen: *Interaction Design – Beyond Human-Computer Interaction*. Wiley and Sons, 2002: S. 41-51

Sehr interessant ist in diesem Zusammenhang die Funktion von konzeptionellen Modellen in Kombination mit speziellen Interfacemetaphern. Hierdurch wird versucht Modelle zu entwerfen, deren Design sich in Bezug auf die Interaktion, an realen physischen Objekten orientiert, aber dennoch gewisse außergewöhnliche Eigenschaften aufweist. Maßgeblich ist hierbei, dass in der Verknüpfung von Interfacemetaphern und konzeptionellen Modellen vorhandene, etablierte Kenntnisse mit neuen Konzepten ergänzt und durch diese modifiziert werden.

Preece führt hierbei u.a. Desktop-Systeme, Suchmaschinen bzw. Formularsysteme als interessante Beispiele an. Verzeichnisse, Dokumente, Mailboxen werden als digitale Elemente mit ähnlichen Eigenschaften wie ihre physisch-realen Pendanten gesehen. Ein Schriftstück am Desktop von einem Verzeichnis ausgehend in ein anderes zu verschieben, ist äquivalent damit, es in einem bestimmten Aktenschrank zu platzieren.

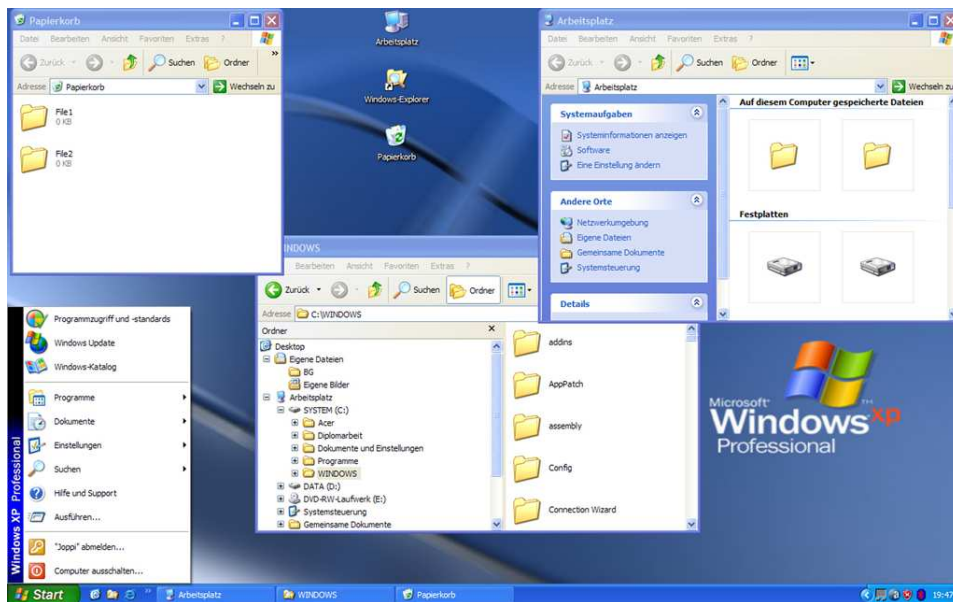


Abbildung 2.10: Beispiel für ein Desktop-System an Hand von Microsoft Windows XP ®.

Grundsätzlich ist zu sagen, dass sich der Einsatz von metaphorischen Darstellungen als sehr erfolgreich erwiesen hat, um den Benutzern das Verständnis für ein interaktives System zu vermitteln bzw. Unterstützung zu bieten, den Umgang mit diesem zu erlernen. Durch die Verwendung von Symbolen, Zeichen und Synonymen aus der physischen, realen Welt, wird den Anwendern ein möglichst intuitiver Weg geboten, um sich in kurzer Zeit in die Materie einzuarbeiten.

Im Kontext der authentisch-realistischen Implementierung ergeben sich aber auch Schwierigkeiten und Herausforderungen. Viele Designer streben rein danach Interfacemetaphern zu entwerfen, die quasi eine Kopie ihres physischen Vorbildes darstellen. Allerdings wird hierbei die eigentliche Absicht verfehlt. Ziel sollte es sein, metaphorische Darstellungen zu kreieren, die vorhandenes auf neues Wissen abbilden, die den Benutzern helfen, durch konventionelle, intuitive Methoden neue Wissensdomänen zu explorieren. Bei allen positiven Argumenten für den Einsatz von Interfacemetaphern führt Jenny Preece auch diverse Kritikpunkte an, die auszugsweise hier erwähnt werden, um die Herausforderungen des Interaktionsdesigns auf Basis dieser Methoden zu illustrieren.¹⁷

The Interface Metaphor:

- *breaks the rules*
Ein signifikanter Kritikpunkt sind die kulturell bedingten, unterschiedlichen Auffassungen und damit verbundenen Widersprüchlichkeiten bei der Interpretation von Metaphern. Oft müssen bewusst Regeln gebrochen werden, um eine allgemeingültige Repräsentation zu generieren.
- *conflicts with design principles*
Viele Interfacemetaphern sind entworfen, um den Bedingungen der physischen Welt zu genügen. Dadurch sind oftmals auch Designer gezwungen, ihre Arbeit diesen Gesetzmäßigkeiten anzupassen. Diese Vorgaben treten dabei häufig auch in Konflikt mit Grundkonzepten der Gestaltung. Dies kann sich negativ auf die Qualität des Designs auswirken.
- *not being able to understand the system functionality beyond*
Benutzer sind oft sehr stark darauf fixiert, ein Verständnis für das System zu entwickeln, basierend auf der metaphorischen Abbildung. Dadurch wird es schwierig zu erkennen, welche zusätzlichen Optionen das Interface zu bieten hat.
- *limits the designer's imagination*
Für Designer besteht eine große Gefahr darin, sich auf etablierte Ideen, gestützt von bekannten Technologien, zu versteifen, um sicherzugehen, dass die Benutzer die entwickelten Methoden auch verstehen und anwenden können.

¹⁷Vgl. [42]: Preece, 2002: S. 57

Die hier angeführten Punkte sollen Designern als Anstoß dienen, Konzepte zu kreieren, die nicht nur den Konventionen der metaphorischen Interfacegestaltung genügen, sondern die vor allem an gewissen Grundprinzipien orientiert sind, die das Fundament für Ideen und Ansätze darstellen, die neue Perspektiven eröffnen.

2.3.2 Varianten von Interaktionsparadigmen

Jenny Preece bezeichnet mit dem Begriff des Interaktionsparadigmas einen bestimmten Weg bzw. eine bestimmte Philosophie des Arbeitens in Bezug auf Interaktionsdesign. Er dient zur Orientierung für die Designer innerhalb ihres Arbeitsablaufes und unterstützt sie dabei, die richtigen Fragen zu stellen und nach den passenden Antworten zu suchen.

Das vorherrschende Interaktionsparadigma war laut Preece für viele Jahre das Prinzip der Desktop-Systeme, die dem typischen und gängigen Bild der Mensch-Computer-Kommunikation (Mensch, CPU, Bildschirm, Mouse, Tastatur) entsprechen. Hierbei ist auch bedeutend, dass Software-Applikationen, die für diese Variante entwickelt wurden, mehrheitlich auf der Verwendung von typischen grafischen Benutzeroberflächen (*GUI*) bzw. *WIMP-Interfaces* (*Windows, Icons, Menus and Pointers*, alternativ auch bezeichnet als *Windows, Icons, Mouse and Pull-Down-Menus*¹⁸) basieren. Durch den zunehmenden Einsatz von neuen Technologien, speziell im Bereich *Wireless-, Mobile- und Handheld-Technologies*, geht der Trend immer mehr in jene Richtung, Paradigmen zu entwickeln, die sich nicht mehr nur auf die gewohnten Prinzipien der Desktop-Systeme stützen. Eine Vielzahl von neuen Interaktionsparadigmen ist von den Wissenschaftlern und Forschern auf diesem Gebiet erdacht, konzipiert und zu einem gewissen Teil auch umgesetzt worden. Preece gibt hierzu einige Beispiele die dafür gedacht sind, eine neue Basis für das System-Development zu bilden und das zukünftige Interaktionsdesign zu prägen.

Beispiele für Interaktionsparadigmen nach Preece:¹⁹

1. Ubiquitous Computing (UbiComp)

Zur Klärung dieses Begriffes verweist Preece auf die Definitionen von Mark Weiser, einem Vorreiter auf dem Gebiet der *HCI*. In seinen Visionen vom ubiquitären Einsatz von Computern verschwinden diese vollständig aus dem unmittelbaren Blickfeld, sodass sich der Benutzer rein auf die Auseinandersetzung mit seinem Vorhaben beschränken kann. Es sollte quasi möglich werden, Computer zu benutzen ohne über sie nachzudenken. Bei der Mehrheit der momentan existierenden Multimediaapplikationen ist dies aber nicht der Fall. Weiser sieht den

¹⁸Siehe [42]: Preece, 2002: S. 60

¹⁹Vgl. [42]: Preece, 2002: S. 62

Anwender gefangen in einer mentalen Sackgasse, in der sich die gesamte Aufmerksamkeit nur noch auf multimediale Repräsentationen, das Bewegen in virtuellen Räumen und das Manipulieren von virtuellen Objekten beschränkt. Nach seinen Ansichten sollten Computer nahtlos in die physische Welt integriert werden und das auf eine Art und Weise, die den menschlichen Fähigkeiten und Kompetenzen entspricht.

Mit folgendem Zitat sollen seine Ansichten verdeutlicht werden:²⁰

Ubiquitous computing will produce nothing fundamentally new, but by making everything faster and easier to do, with less strain and fewer mental gymnastics, it will transform what is apparently possible.

2. Pervasive Computing

Diese Form des Einsatzes von Computern kann als Ansammlung von Nachfolgeideen bezogen auf *Ubiquitous Computing* gesehen werden. Die Grundidee dahinter besagt, dass die Benutzer zu jeder Zeit an jedem beliebigen Ort in der Lage sein sollen, beliebige Technologien anzuwenden ohne spezielle Integrationsmethoden einzusetzen. Mehrheitlich werden mit dieser Methodik *Smart Devices* in Verbindung gebracht, die in erster Linie Informationsdienste präsentieren.

Vielmehr geht es aber auch darum, Objekte des allgemeinen, täglichen Gebrauchs zu integrieren. Als Beispiele lassen sich z.B. intelligente Kühlschränke nennen, die einen Lebensmittelengpass signalisieren oder Mikrowellen, die Informationen aus dem Internet während des Kochvorganges abrufen.

3. Tangible Bits, Augmented Reality and Virtual Integration

Eine weitere Entwicklung, die ausgehend von *Ubiquitous Computing* stattgefunden hat, ist im Bereich der *Tangible User-Interfaces* zu erkennen. Ishii und Ulmer sprechen in diesem Zusammenhang in [28] von so genannten Tangible Bits²¹ (vgl. Abschnitt 2.6.3).

Der Fokus liegt bei diesem Interaktionsparadigma auf der Kombination von digitalen Informationen mit physischen Repräsentationen, um den Benutzern erweiterte Möglichkeiten in der Durchführung ihrer alltäglichen Tätigkeiten zu bieten. Beispiele hierfür sind u.a. Bücher, die digitale Hinweise beinhalten, Postkarten, die Animationen abspielen und vor allem auch physische Objekte, die mit virtuellen in Verbin-

²⁰Siehe: Weiser Mark: *The Computer for the 21st Century*. Scientific American, 265 (3), 1991: S. 94-104 – In [42]: Preece 2002: S. 62

²¹Siehe [28]: Ishii Hiroshi, Ullmer Brygg: *Tangible Bits – Towards Seamless Interfaces between People, Bits and Atoms*. MIT Media Laboratory, 1997: S. 1

dung stehen und Effekte der Interaktion auf diese übertragen. Preece führt zusätzlich als quasi Erweiterung dieses Paradigmas das Prinzip der *Augmented Realities* an, das in vorangegangenen Abschnitten dieser Arbeit schon detailliert erläutert worden ist (siehe Abschnitt 2.1.3). Grundsätzlich bleibt festzuhalten, dass die Forschung vorangetrieben wird, um die Kluft zwischen der realen und virtuellen Welt zu verringern. Es werden z.B. im Officebereich permanent Tools entwickelt, die physische Eigenschaften mit virtuellen Repräsentationen verbinden. Ziel ist es, einen nahtlosen Übergang zwischen beiden Welten zu finden.

4. Attentive Environments and Transparent Computing

Innerhalb dieses Interaktionsparadigmas nimmt der Computer die Rolle eines Kommunikationspartners ein, der die Bedürfnisse und Wünsche der Benutzer vorausahnt. Diese müssen sich nun keine Gedanken mehr darüber machen, welchen Arbeitsschritt sie als nächstes durchführen, bzw. welche Methode sie anwenden, um ihr Ziel zu erreichen. Sämtliche dieser Aufgaben sollen vom Computer übernommen werden, was natürlich entsprechende Anforderungen an das System stellt. Es muss in der Lage sein, den aktuellen Zustand des Anwenders, sowie seine Ansprüche und Wünsche zu erkennen. Hierzu werden Kameras und Mikrofone eingesetzt, um das Verhalten an Hand von Gestik, Ausdruck und akustischen Signalen zu analysieren. Mit den erhaltenen Daten wird ein Profil erstellt, das den physischen, emotionalen und informationellen Status beinhaltet und woraus nun abgeleitet werden kann, welches Verhalten zu erwarten ist bzw. welche Informationen für den Benutzer interessant sein könnten.

Die grundsätzliche Form der Interaktion tritt hierbei in den Hintergrund, da das Hauptaugenmerk auf der Zustandsanalyse liegt.

5. The Workaday World

Bei den vorangegangenen Interaktionsparadigmen lag der Schwerpunkt mehrheitlich auf der Erforschung, wie digitale Informationen mit physischen bzw. virtuellen Objekten gekoppelt werden können, um den Benutzern Möglichkeiten zu bieten, die sie zuvor noch nicht hatten. Im Gegensatz dazu führt Preece das so genannte *Workaday-Paradigma*²² an, das von Tom Moran und Bob Anderson spezifiziert wurde. Im Rahmen ihrer Forschungsarbeit setzten sie sich mehrheitlich mit den sozialen Aspekten von Technologie auseinander, speziell in Hinblick auf die Nutzbarkeit für Designer. Sie fokussierten sich dabei in erster Linie auf den Begriff des Arbeitsplatzes in Kombination mit Wissen,

²²Vgl.: Moran Tom, Anderson Bob: *The Workaday World as a paradigm for CSCW Design*. (In: Proceedings of the CSCW 90, 381-393) – In: [42]: Preece 2002: S. 64

Produktionsmitteln und alltäglichen Tätigkeiten der Benutzer. Ihre Arbeit stellt in diesem Zusammenhang einen wichtigen Beitrag zur Differenzierung der verschiedenen Settings dar, in denen Technologie existiert und angewendet wird.

2.3.3 Zusammenfassung

Die Konzepte und Theorien die Jenny Preece präsentiert, um eine Definition des Begriffes Interaktionsparadigma zu liefern, stellen eine interessante Alternative zu den bisher besprochenen Varianten dar. Sie sieht Interaktionsdesign an sich als iterativen Prozess an, in dem verschiedene Aufgabenstufen durchschritten werden. Entscheidend ist für sie in diesem Zusammenhang vor allem, die Ansprüche der Anwender zu verstehen und zu extrahieren und sie auf Basis adäquater konzeptioneller Modelle in die Realität umzusetzen. Konkrete Punkte, die es dabei zu spezifizieren gilt, sind z.B. die Möglichkeiten die ein interaktives System bieten muss, wie es sich verhält und speziell welche Varianten der Interaktion ausgewählt werden, um den Benutzer in den Aktionsraum des Systems zu integrieren.

Im Kontext dieser Thematik spricht sie auch von den Zusammenhängen zwischen konzeptionellen Modellen und spezifischen Interfacemetaphern. Hierbei wird versucht Konzepte der Interaktion zu kreieren, die sich am Umgang mit realen physischen Objekten orientieren. Ziel sollte es dabei sein, vorhandenes Wissen der Benutzer zu verwenden, um ihnen die Möglichkeit zu bieten, sich auf eine intuitive Art und Weise mit dem System auseinander zu setzen und neue Erkenntnisse zu gewinnen.

Der Begriff des Interaktionsparadigmas definiert für Preece grundsätzlich eine bestimmte Form des Denkens in Bezug auf Interaktionsdesign. Für Designer soll er eine Unterstützung darstellen und Quelle der Inspiration sein. Das vorherrschende Paradigma war für sehr lange Zeit das Prinzip der Desktop-Systeme. Dieses wurde aber durch den vermehrten Einsatz von neuen Technologien mehr und mehr in den Hintergrund gedrängt. Eine Vielzahl von neuen Interaktionsparadigmen sind bereits konzipiert und umgesetzt worden. Sie stellen quasi eine neue Basis für die Entwicklung von interaktiven Systemen dar und prägen die Tätigkeit der Systementwicklung maßgeblich.

Im Rahmen dieser Arbeit wurden folgende Interaktionsparadigmen nach Preece präsentiert:

- Ubiquitous Computing
- Pervasive Computing
- Tangible Bits, Augmented Reality and Virtual Integration
- Attentive Environments and Transparent Computing
- The Workaday World

Grundsätzlich bleibt zu sagen, dass die von Preece spezifizierten Konzepte Varianten der Kommunikation zwischen Mensch und Computer darstellen. Dabei geht es weniger darum, wie nun Daten ein- bzw. ausgegeben werden oder wie Informationen strukturiert und präsentiert werden. Vielmehr ist essentiell, welche Methoden der Benutzer anwenden kann, um mit dem System zu kommunizieren. Festzuhalten bleibt, dass die angeführten Konzepten und technischen Varianten wiederum nur Auszüge der multiplen Möglichkeiten darstellen, die den Designern und Entwicklern im Rahmen ihrer Arbeit zur Verfügung stehen.

2.4 Interfaceparadigmen nach Jef Raskin

We teach civility in daily personal interaction, we must teach it in computer interaction as well. (Jef Raskin)

Jef Raskin gehört unumstritten zu den revolutionären Vorreitern auf dem Gebiet der *HCI* und dies nicht zuletzt durch seine Mitarbeit in zahlreichen Forschungsprojekten. Zu seinen herausragenden Leistungen zählt die Kreation und Leitung des Macintosh-Projektes bei *Apple*, wodurch eine außergewöhnliche Technik der Kommunikation zwischen Mensch und Computer geschaffen wurde. Der Einsatz grafischer Benutzeroberflächen stellt einen wesentlichen Meilenstein in der Entwicklung von Computer-Systemen dar. Auf dieser Basis aufbauend wurden Konzepte, Methoden und Techniken etabliert, die heute als universal angesehen werden.

Raskin hatte zudem einen Lehrstuhl für *Computer Science* an der Universität Chicago inne und unterrichtete an den Universitäten von Californien und Stanford. Im Rahmen seiner wissenschaftlichen Tätigkeit veröffentlichte er neben mehreren Büchern zahlreiche Artikel zur Thematik. Für Jakob Nielsen zählt er in einer Zeit in der die meisten damit zufrieden sind, Designs und Konzepte zu kopieren, die längst an Bedeutung verloren haben, zu den Koryphäen der *HCI*.²³

Raskin sah die Zeit der kompromisslosen Auseinandersetzung mit der digitalen Technologie als beendet an. Die Benutzer sind nicht mehr gewillt, eine Vielzahl von Bedienungsschritten zu erlernen, um auch nur die einfachsten Aufgaben durchzuführen. Vielmehr geht die Entwicklung in jene Richtung, intuitive Interfaces zu entwickeln, die in der Lage sind, die Anwender zu animieren, die Komplexität ihrer Aufgaben zu steigern, anstatt ihre Zeit mit dem Erlernen simpler Methoden der Handhabung zu verschwenden.

Raskin weist auf viele Methoden und Konzepte der Interaktion hin, die nicht mehr den heutigen Anforderungen der Computernutzung gerecht werden. Er spricht in diesem Zusammenhang von Interfaceparadigmen, die in

²³Vgl. [46]: Raskin Jef: *The Humane Interface – New Directions for Designing Interactive Systems*. Addison-Wesley, 2005: S. 233

einer Sackgasse angelangt sind. Zusätzlich versucht er aber auch Erklärungen und Ideen zu liefern, welche Möglichkeiten für Designer und Entwickler potentiell zur Verfügung stehen. Die von ihm definierten Konzepte sind dabei von wesentlicher Bedeutung, da sie seine Erfahrung auf wissenschaftlich-theoretischer Ebene, sowie auch auf der praktischen Ausführungsebene widerspiegeln.

2.4.1 Interaktionsanalyse im Kontext der bewussten bzw. unbewussten Wahrnehmung

Jef Raskin beschäftigt sich im Rahmen seiner wissenschaftlichen Auseinandersetzung mit den Prinzipien der Mensch-Computer-Kommunikation, sehr intensiv mit psychologischen und physiologischen Aspekten der menschlichen Wahrnehmung. Für ihn ist es ein entscheidendes Faktum, die mentalen Fähigkeiten des Menschen genau definieren zu können, da diese im wechselseitigen Arbeitsverhältnis zwischen Mensch und Maschine sehr stark beansprucht werden. Er betrachtet diese Begriffe hierbei mehrheitlich im Kontext der Interaktion, als Form der Kommunikation, als Prozess, in dem die Handlung von bewussten bzw. unbewussten Wahrnehmungen geprägt ist. Diese Zusammenhänge zu kennen ist entscheidend, will man in der Lage sein, den Benutzer durch den Aktionsraum eines interaktiven Systems zu lenken. Raskin prägt in diesem Zusammenhang den Begriff des *Locus of Attention*²⁴, wobei er sich auf die Zentren der menschlichen Aufmerksamkeit bezieht und hierbei vor allem auf die Fähigkeit, sich auf bestimmte Objekte bzw. Handlungen zu fokussieren.

Wenn der Benutzer angehalten ist, einen komplexen Handlungsablauf mehrmals durchzuführen, so wird ihm dies durch die andauernde Wiederholung bei fortschreitender Zeit leichter fallen. Die vormals komplexe Aufgabenstellung wird zu einer gewohnten Situation und die Ausführung wird mehr und mehr automatisiert. Dadurch wird bewirkt, dass man nicht mehr aktiv über die Bewältigung an sich nachdenkt, sondern nur noch handelt.

Im Zusammenhang mit Interaktionsdesign sieht es Raskin als obligat an, Interaktion zu kreieren, die auf bestimmten Konzepten basiert, die zum einen für die Benutzer leicht zu erkennen sind und zum anderen keine Barrieren bzgl. des Zuganges zum System darstellen können. Aus diesem Grund ist es entscheidend, ein kontinuierliches Design zu entwickeln, um den *Locus of Attention* auf das Lösen der gestellten Aufgabe zu lenken und nicht auf das Erlernen von komplexen Interaktionsmodi. Im folgenden Zitat beschreibt Raskin diesen Sachverhalt sehr deutlich:²⁵

The ideal humane interface would reduce the interface components of a users's work to begin habituation. Many of the problems

²⁴Siehe [46]: Raskin, 2005: S. 17

²⁵Siehe [46]: Raskin, 2005: S. 20

that make products difficult and unpleasant to use are caused by human-machine design that fails to take into account the helpful and injuries properties of habit formation.

Er spricht in diesem Kontext auch von der Ausbildung von Automatismen, die es erst ermöglichen, mehrere Aktionen zur selben Zeit durchzuführen. Denn umso unbewusster und automatisierter eine Tätigkeit abgewickelt werden kann, desto mehr Spielraum bleibt für parallele Aufgaben, wodurch wiederum die Effizienz gesteigert werden kann.

Ausgehend von den Erkenntnissen und Annahmen, die Raskin durch seine eingehende Beschäftigung mit den psychologischen Aspekten der Mensch-Computer-Kommunikation gewonnen hat, führt er grundsätzliche Methoden zur Analyse und Bewertung von Interface- bzw. Interaktionskonzepten an. Im Zuge der Definition des Begriffes Interaktionsparadigma, erweisen sich diese Konzepte als sehr hilfreich, da sie weitere Möglichkeiten der Klassifizierung bieten. Hierbei bestehen auch gewisse Analogien zu den Theorien von Shneiderman und Preece aus den vorangegangenen Kapiteln.

Vorrangig spricht Raskin von so genannten *Interface Timings*²⁶ zur Messung der Interfaceeffektivität. Grundsätzlich wird hierbei versucht die Zeit zu ermitteln, die der Computer benötigt, um einen Befehl des Benutzers auszuführen. Eine Anweisung wird dabei als Summe der seriellen, einzelnen Befehlsschritte gesehen, die nach dem Schema Befehl - Response (Antwort) - Ausführung - Feedback ablaufen. Dadurch ergibt sich die grundsätzliche Möglichkeit, eine Unterteilung von Interaktionsdevices bzw. Interfaces an Hand der Interaktionsgeschwindigkeit vorzunehmen. Raskin prägt in diesem Zusammenhang den Begriff *Information-Theoretic-Efficiency*²⁷ mit dessen Hilfe es möglich ist, quasi den Wirkungsgrad eines interaktiven Systems zu ermitteln. Dieser definiert sich dabei als die minimale Informationseinheit, die benötigt wird, um eine Aufgabe durchzuführen, dividiert durch den Grad der Information den der Benutzer dafür aufwendet.

Darin liegt aber wiederum auch die große Schwierigkeit dieses vereinfachten Modells, da sich die Interpretation der gemessenen Werte nicht einfach gestaltet. Im Speziellen bei der Gegenüberstellung und Analyse von komplexen Interaktionskonzepten lässt sich die Performance nur sehr schwer durch Zahlen festhalten. Hinzu kommt, dass die Ergebnisse natürlich stark von den individuellen Fähigkeiten der Probanden abhängig sind. Allerdings eignen sich solche Vergleiche sehr gut, um die positiven bzw. negativen Aspekte des Interfacedesigns zu evaluieren.

²⁶Siehe [46]: Raskin, 2005: S. 73

²⁷Siehe [46]: Raskin, 2005: S. 85

2.4.2 Intuitive und Natürliche Interaktion: Varianten von Interfaceparadigmen I

Natürlichkeit und Intuition sind im Interaktionsdesign zu Grundanforderungen avanciert, wenn es um das Design von interaktiven Systemen geht. Berechtigterweise darf man aber hier die Frage stellen, was unter einem natürlichen bzw. intuitiven Interface zu verstehen ist. Oftmals verbinden die Benutzer damit eine Arbeitspraxis, die sie aus ähnlichen Softwareapplikationen bzw. verwandten Tätigkeiten kennen. Mehrheitlich bezeichnet es also eine Bedienoberfläche, die man gewohnt ist und sich dadurch sehr schnell orientieren kann, um sich der eigentlichen Tätigkeit zu widmen. Der Erfolg eines Benutzerkonzeptes lässt sich nun also weniger dadurch erreichen, dass man es intuitiv oder natürlich gestaltet, sondern vielmehr durch den Einsatz von etablierten und wohl bekannten Interaktionstechniken. Raskin verweist hierbei auf die zunehmende Vernetzung zwischen unterschiedlichen Informationsquellen, wodurch sich die Anforderungen an *Wissensportale* bzw. *Information Spaces* erheblich gesteigert haben. In diesem Kontext ist es eine große Herausforderung, den Zugang zu gewährleisten durch adäquate Visualisierungen der Zusammenhänge.

Raskin prägt hierbei den Begriff des *Zooming Interface Paradigms (ZIP)*²⁸, das zum einen Problemen der Navigation entgegenwirkt und zum anderen eine Lösung für die Visualisierung von Inhalten auf größenlimitierten Benutzeroberflächen beschreibt. Obwohl das *ZIP* natürlich nicht immer optimal für alle Situationen geeignet ist, bietet es dennoch speziell für die Gestaltung der Benutzerschnittstelle im Rahmen des Interaktionsdesigns neue Möglichkeiten. Information und interaktive Elemente sind nun auf einer beliebig großen Oberfläche verteilt, auf der man frei navigieren kann. Durch das *Zooming* ist es möglich, in die verschiedenen Hierarchien einzutauchen und unterschiedliche Informationsebenen zu explorieren. Teilbereiche, die zuvor nur unschwer zu erkennen waren, können nun im Detail betrachtet werden. Raskin sieht das Konzept des *ZIP* als Erweiterung zu typisch Desktop-orientierten Applikationen, die in ihrer Ausbreitung doch eher eingeschränkt sind und vergleicht das Interaktionsprinzip mit einem Blick aus der Vogelperspektive auf einen Irrgarten, wodurch sofort sämtliche Wege klar vor Augen liegen.

Das Arbeiten mit einem *ZIP* verlangt Mehrheitlich das Ablegen der typischen Denkmuster der Interaktion, wie man sie aus den herkömmlichen Desktop-basierten, grafischen Benutzerschnittstellen kennt. Dies liegt in erster Linie daran, dass man nun nicht mehr in einer klar begrenzten Ebene interagiert, sondern eine Struktur vor sich hat, die bis zu einem gewissen Maß keinen räumlichen Grenzen gegenübersteht.

Interessant sind in diesem Zusammenhang auch die Möglichkeiten des kollaborativen Zusammenarbeitens, die geboten werden. Ein Datennetzwerk

²⁸Siehe [46]: Raskin, 2005: S. 153

kann repräsentiert werden als *ZIP-Space*²⁹, in dem jeder Partizipierende seinen eigenen Bereich besitzt, der verwaltet werden kann. Informationen können nun beliebig zugänglich gemacht und untereinander ausgetauscht werden. Unsichtbare Datensätze können dabei auch eine Form der Zugangskontrolle darstellen.

Wie also unschwer zu erkennen ist, kann das *ZIP* eine interessante Alternative zu den herkömmlichen Desktop-Interaktionsparadigmen darstellen. Das Prinzip ist dabei auf verschiedenste Art und Weise anwendbar, ob beim Web-Browsing oder auch bei der Dateiverwaltung.

2.4.3 Techniques in Humane Interfaces: Varianten von Interfaceparadigmen II

Raskin sieht eines der Hauptziele des Interaktionsdesigns darin, Interfaces zu kreieren, die mehr oder weniger selbsterklärend zu bedienen sind. Dies wird dabei weniger durch die Integration von Intuition bzw. Natürlichkeit in das System erreicht, sondern vielmehr durch ein durchdachtes Netz von Hinweisen, Instruktionen und Erklärungen. Dies setzt vor allem voraus, dass sämtliche Bereiche der Benutzerschnittstelle jederzeit erreichbar sind, speziell was Referenz-Manuals bzw. Hilfethemen betrifft. Die Herausforderung besteht dabei vor allem darin, diese Elemente ohne spezielle Zugangsmechanismen oder -techniken nahtlos in das Interface zu integrieren – sie müssen ein Teil des Ganzen werden.

Im vorangegangenen Abschnitt wurde aufgezeigt, dass so etwas durchaus möglich ist. Das Konzept der *Zooming Interfaces* definiert eine Arbeitsoberfläche, in der einschränkende Elemente auf ein Minimum reduziert sind. Verzeichnisstrukturen werden vollständig eliminiert, da nur mehr eine universale Oberfläche existiert, auf der die Datenobjekte in Datenblöcken organisiert sind. Diese Art und Weise der Implementierung des Interfaces verlangt natürlich auch nach speziellen Formen der Interaktion. Raskin spezifiziert in diesem Zusammenhang Formen von Interfaceparadigmen, die in einem solchen Umfeld Anwendung finden können und nachfolgend beschrieben werden.

Innerhalb des *Zoom-Space* werden einzelne, zu bearbeitende Objekte nicht mehr einfach nur geöffnet oder geschlossen, sondern durch den Vorgang des *Zoom In* bzw. *Zoom Out*³⁰ vergrößert oder verkleinert und erfahren damit eine Translation in den Wirkungsraum des Systems. Ein entscheidender Aspekt in Bezug auf *cut and paste* ist hierbei, dass Inhalte nicht mehr von Null ausgehend auf Basis einer blanken, leeren Oberfläche erzeugt werden, sondern dass vielmehr bereits bestehende Elemente, in Kombination mit den Tätigkeiten des Ausschneidens (*cut*) und Einfügens (*paste*), die Grundlage für neue Arbeitsobjekte darstellen.

²⁹Siehe [46]: Raskin, 2005: S. 156

³⁰Siehe [46]: Raskin, 2005: S. 175

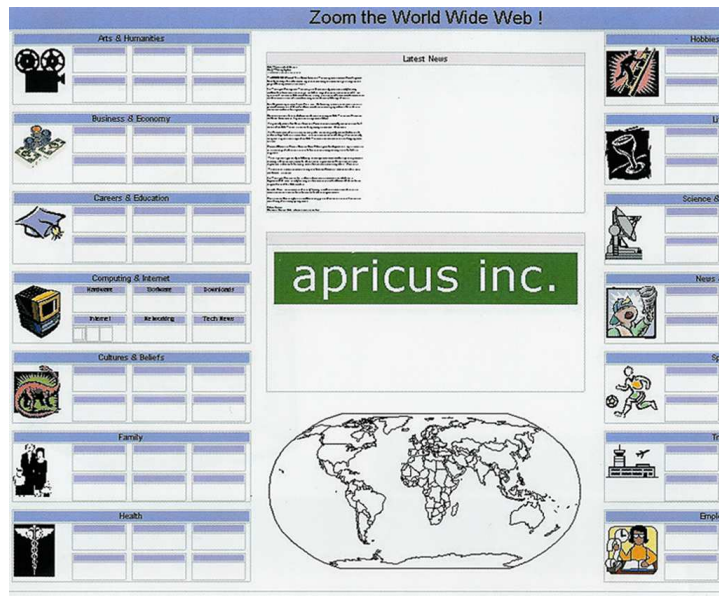


Abbildung 2.11: Implementierung des *Zooming Interface Paradigm* im WWW – Informationen sind im *Zoom Space* in verschiedenen Datenblöcken organisiert. Aus [46].

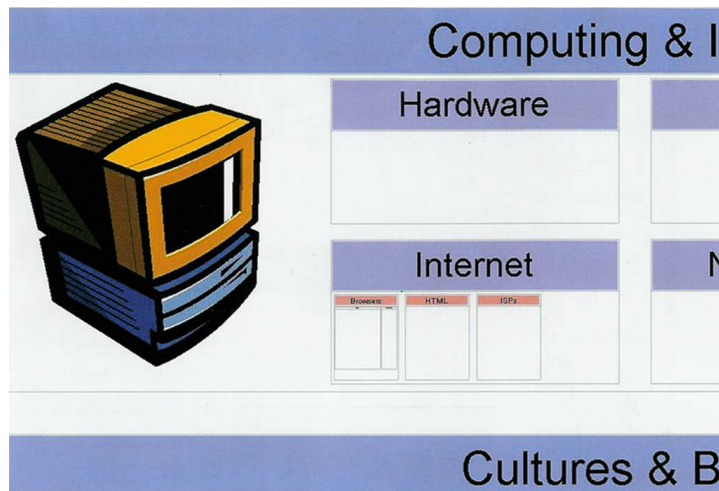


Abbildung 2.12: Implementierung des *Zooming Interface Paradigm* im WWW – Interaktion im *Zoom Space* durch den Vorgang des *Zoom In*. Aus [46].

In diesem Zusammenhang lassen sich Parallelen zu Jakob Nielsen ziehen, der ähnliche Theorien zur Thematik konstatierte. Er spricht in Nielsen's Alertbox vom 10. Oktober 2005³¹ vom Interaktionsparadigma der so genannten *Result-Oriented User Interfaces*, die einen neuen Weg der Mensch-Computer-Interaktion ebnen. Ähnlich wie beim *cut and paste* Paradigma von Raskin, haben die Benutzer dabei die Möglichkeit, aus bereits bestehenden Elementen und Inhalten neue zu erstellen.

2.4.4 Zusammenfassung

Jef Raskin hat in seiner langjährigen Tätigkeit im Bereich der Mensch-Computer-Kommunikation in zahlreichen Forschungsprojekten mitpartizipiert, in denen Konzepte und Techniken der Interaktion entwickelt wurden, die heute als universal gelten. Entscheidend war in diesem Zusammenhang, dass er seine wissenschaftlichen Erkenntnisse auch in der Praxis, im Rahmen des Macintosh-Projektes bei Apple, umsetzen konnte. Dadurch rückte der Begriff des Interfacedesigns in den Fokus der Designer und Entwickler und erreichte einen bedeutenden Stellenwert.

Raskin vertrat die Ansicht, dass die Zeit der kompromisslosen Auseinandersetzung mit der digitalen Technologie beendet ist. Die Benutzer seien nicht mehr gewillt, für das Durchführen einfachster Aufgabenschritte, komplexe Bedienungsschritte zu erlernen. Das Ziel sei es vielmehr, Benutzerschnittstellen so einfach wie möglich zu gestalten unter Bedachtnahme der Beschränkungen und der potentiellen Möglichkeiten (jeweils vorgegeben durch das System). Es ist entscheidend zu wissen, was die Anwender in der Lage sind zu tun und was nicht. Raskin verweist vorrangig auf die Ergebnisse aus seiner sozialwissenschaftlichen Forschung und hierbei vor allem auf die Grundzüge der bewussten bzw. unbewussten Wahrnehmung. Er spricht in diesem Zusammenhang von einem so genannten *Locus of Attention*, der in gewisser Weise das Zentrum der menschlichen Aufmerksamkeit beschreibt. Viele Interface-Methodologien richten sich aber nicht nach diesen Kriterien und bewirken eine Überforderung der Anwender. Ziel sollte es sein, eine geringe Anzahl von elementaren Interaktionsmöglichkeiten zu definieren, die an eine breite Basis von verschiedensten potentiellen Anwendungsbereichen angepasst werden können. Raskin definiert hierzu das Interfacekonzept der *Zooming Interfaces*. Informationen sind in Datenblöcken organisiert, die auf einer beliebig großen Fläche verteilt sein können, auf der man frei navigiert. Die Tätigkeit des *Zooming* stellte dabei eine interessante Metapher dar, denn dadurch wird es möglich, die verschiedenen Informationsebenen zu erkunden. Teilbereiche, die zuvor nur unschwer zu erkennen waren, können nun im Detail betrachtet werden. Dieses Prinzip verzichtet dabei auf Dateibezeichnungen bzw. starre Strukturen der Verzeichnisorganisation.

³¹Siehe [39]: Nielsen Jakob: *Jakob Nielsen's Alert Box, October 10, 2005: R.I.P. WY-SIWYG*. <http://www.useit.com/alertbox/wysiwyg.html>, 13. Nov.: 23:53, 2006

Ein wesentliches Merkmal für das *ZIP* ist auch die Art und Weise der Erzeugung von Inhalten. Raskin verweist hierbei auf die Methode des *cut and paste*, die dem Benutzer die Möglichkeit bietet, Elemente aus verschiedenen Quellen zu einem neuen Ganzen zu kombinieren. Hierbei lassen sich Parallelen zu den Theorien von Jakob Nielsen über so genannte *Result-Oriented User Interfaces* ziehen, die eine neue Richtung der *HCI* darstellen.

Es bleibt zu sagen, dass Jef Raskin in den Publikationen, die für diese Arbeit herangezogen wurden (Literaturverzeichnis: [45], [46]), sehr hart ins Gericht geht mit den so genannten Grundregeln des Interaktionsdesigns. Viele der etablierten Methoden sind teilweise nicht mehr zeitgemäß und in einer Sackgasse angelangt. Er übt Kritik, liefert aber auch Ideen und Alternativen. Diese sind vor allem deshalb interessant, weil in ihnen eine langjährige wissenschaftliche und praktische Erfahrung zum Tragen kommt.

2.5 Die Interaktionsparadigmen von Jakob Nielsen

Simplicity should be the goal of page design. (Jakob Nielsen)

Jakob Nielsen fungiert als Forscher und Berater in Bereichen der Softwareentwicklung und im Speziellen in der Software-Ergonomie. Bekannt wurde er aber durch seine wissenschaftliche Tätigkeit auf dem Gebiet der Web-Usability. Er hat sich schon in den frühen Anfängen mit den Grundzügen dieses Mediums auseinandergesetzt und avancierte durch seine regelmäßig erscheinenden Artikel zu einem der bekanntesten Kritiker von Websites in Hinblick auf Informationsaufbereitung, Interaktionsmethodologie und Benutzerfreundlichkeit.

Zusätzlich setzt sich Nielsen mit Aspekten des Usability-Engineering auseinander. Ausschlaggebend ist, dass in seine Theorien und Konzepte, seine langjährige wissenschaftliche und praktische Erfahrung mit einfließen. Auf Basis seiner wissenschaftlichen Auseinandersetzung mit der Thematik formulierte er eine Vielzahl von Designrichtlinien, die aber durchaus kritisch betrachtet werden dürfen.

Seine Ansichten werden innerhalb dieser Arbeit erwähnt, weil sie sehr interessante Ansätze zur Verwendung des Begriffes Interaktionsparadigma beinhalten.

2.5.1 Das Ende von WYSIWYG

Zur Publikation wissenschaftlicher Artikel und Beiträge stützt sich Jakob Nielsen mehrheitlich auf seine Online-Plattform. Er spricht dabei in Nielsen's Alertbox vom 10. Oktober 2005 vom so genannten Interaktionsparadigma *What you see is what you get (WYSIWYG)*, welches in den letzten 25 Jahren die Computernutzung maßgeblich geprägt hat. Es stellt im Vergleich

zu den vorhergehenden Interaktionsformen eine bedeutende Steigerung dar, da die Benutzer das Ergebnis ihrer Aktionen durch eine unmittelbare optische bzw. grafische repräsentative Reaktion des Systems überprüfen können. Im Gegensatz zum Eintippen von Kommandos oder Parametern, interagiert man mit visuellen Objekten, die manipulative Eigenschaften besitzen. Was am Screen dargestellt wird, entspricht dabei zu jeder Zeit dem, was man erstellt, konstruiert bzw. gestaltet hat.

Für Nielsen sind die Grenzen dieses Paradigmas aber bei weitem erreicht. Das grafische Anzeigen von Interaktionswerkzeugen funktioniert bei einer geringen Anzahl von Elementen sehr zufrieden stellend. Allerdings werden die Produkte der Softwareentwicklung, bezogen auf ihren Funktionsumfang, immer umfassender, wodurch die Handhabung äußerst komplex ist. Weiters erfordert das Prinzip des *WYSIWYG* ein gewisses Maß an Vorstellungskraft, da man, ausgehend von einem leeren Dokument, sich Schritt für Schritt zum gewünschten Ziel vorarbeiten muss.

Grundsätzlich definiert Nielsen drei elementare Kritikpunkte am Interaktionsparadigma *WYSIWYG*:³²

- Eine Vielzahl von Kommandos und Befehlen, die teilweise nicht verwendet werden und darüber hinaus jeder für sich lokalisiert werden müssen
- Ein hoher Zeitaufwand, um ein leeres Dokument zum gewünschten Ergebnis zu modellieren
- Eine fehlende Benutzerunterstützung während des kompletten Arbeitsablaufes, da vom Ausgangszustand an nur eine Kombination von verschiedensten Kommandos Ziel führend ist.

Nielsen versucht damit zu verdeutlichen, dass die zukünftige Mensch-Computer-Kommunikation vor einem Wandel steht. Er prägt in diesem Zusammenhang den Begriff der *Result-Oriented User Interfaces*³³, die einen neuen Weg der *HCI* ebnen. Sind die Benutzer beim vorhergehend beschriebenen Prinzip noch mehrheitlich auf sich allein gestellt und müssen sich Schritt für Schritt ihrem eigentlichen Ziel entgegenarbeiten, so werden sie nun die Möglichkeit haben, Elemente und Inhalte aus verschiedenen Quellen auszuwählen, um diese zu einem neuen Ganzen zu kombinieren, das ihren Vorstellungen entspricht. Nielsen verweist dabei auf die Gegebenheit, dass die Mehrheit der Benutzer eher in der Lage ist, ein bereits bestehendes Design zu modifizieren, als ein neues von Null auf zu kreieren. Das Interaktionsparadigma des *WYSIWYG* wurde damit umgekehrt zu *What you get is what you see*, kurz *WYGIWYS*.³⁴ Nielsen führt hierbei eine interessante

³²Vgl. [39]: Nielsen, 2006

³³Siehe [39]: Nielsen, 2006

³⁴Siehe [39]: Nielsen, 2006

Metapher an und vergleicht das Prinzip von *WYGIWYS* mit dem Schaffen einer Steinskulptur. Hierbei beginnt man aber nicht mehr ausgehend von einem blanken unförmigen Steinblock, sondern verwendet eine bereits fertige Statue und führt beliebige Modifikationen durch.

Ein Hauptargument, das zur Etablierung dieses Interaktionsparadigmas beitragen kann, ist der Aspekt der Produktivitätssteigerung. Die Benutzer sind nun in relativ kurzer Zeit in der Lage, ein gewisses Basisergebnis zu erzeugen und sind nicht gezwungen, eine Vielzahl von Befehlsketten durchzuarbeiten. Nielsen sieht den vorrangigen Einsatzbereich dabei in verschiedensten Varianten von Office-Anwendungen. Entscheidend wird aber sein, ob und wie sich dieses neue Interaktionsparadigma in andere Softwaregebiete übertragen lassen wird.

2.6 Varianten und Formen der direkten Interaktion: Multimodalität

There is a great satisfaction in building good tools for other people to use (Freeman Dyson)

Der Begriff des Interaktionsparadigmas wird in der Literatur und auch innerhalb dieser Arbeit mehrheitlich verwendet, um in gewisser Weise System- bzw. Kommunikationsmodelle zu beschreiben. Wie innerhalb dieser Modelle die Interaktion an sich realisiert wird, ist weniger Bestandteil der Betrachtung. Um nun aber ein multimediales System kreieren zu können, ist es unbedingt notwendig, auf die unterschiedlichen Formen der direkten Interaktion einzugehen und darzustellen, wie nun die direkte Kommunikation zwischen Mensch und Computer realisiert werden kann. Wichtig ist es hierbei zu definieren, auf welche Art und Weise der Benutzer eine angestrebte Aktion durchführen und wie Daten-Input bzw. -Output generiert werden kann. Die Interaktion kann dabei z.B. über eine graphische Benutzeroberfläche in Kombination mit einer Mouse als Manipulationswerkzeug erfolgen oder auch durch den Einsatz natürlicher Sprache. Die Varianten und Formen sind sehr vielfältig und unterschiedlich, wodurch sich multiple Möglichkeiten der Gestaltung bieten.

Im Folgenden werden wiederum Beispiele präsentiert, um einen Eindruck zu vermitteln, was nun als Interaktionsform gesehen werden kann. Diese Angaben sollen nun keine zusätzliche Definition des Begriffes Interaktionsparadigma darstellen, sondern lediglich als Ergänzung zur bereits detailliert besprochenen Thematik fungieren.

2.6.1 Graphical User Interfaces – GUIs

Eine grafische Benutzeroberfläche (Englisch: Graphical User Interface – *GUI*) ist eine spezielle, deskriptive Form der Interaktionsschnittstelle zwi-

schen Mensch und Computer. Grundlegender Aspekt ist hierbei, die Bedienung von Software-Applikationen einfacher und handlicher zu gestalten. Verschiedene Elemente der Benutzeroberfläche, wie z.B. Operatoren, Werkzeuge oder ganze Befehlsketten, werden durch entsprechende grafisch-metaphorische Abbildungen (*Symbole, Piktogramme*) repräsentiert. Beispiele für solche Interaktionskomponenten sind u.a. *Windows, Buttons, Menus, Scroll-Bars, etc.*³⁵ Im Laufe der Zeit entwickelte sich daraus der eigenständige Begriff der so genannten *WIMP-Systeme* (*Windows, Icons, Menus, Pointers*). Dieser beschreibt einen speziellen Aufbau der grafischen Benutzeroberfläche mit dezidierten, essentiellen Komponenten und wird im Zusammenhang mit *GUIs* verwendet.

Das System der grafischen Benutzeroberflächen erfreut sich großer Popularität und findet Anwendung in sehr vielen verschiedenen Softwareapplikationen. Die Vorteile davon liegen klar auf der Hand:

- Die symbolbehaftete Sprache stellt eine Kommunikationsbasis für verschiedenste Benutzer dar
- Komplexe Operationen bzw. Befehlsketten lassen sich durch einfache repräsentative Darstellungen illustrieren.
- Zwischen dem Symbol und der repräsentierten Aktion besteht ein sichtbarer Zusammenhang

In diesem Kontext ist es sehr wichtig anzuführen, dass eine grafische Benutzeroberfläche nur dann Sinn macht, wenn sie ordentlich gestaltet ist. Die bildhafte Darstellung der Interaktionskomponenten verbraucht viel Platz, wodurch das Interface schnell überladen wirken kann und in weiterer Folge die Orientierung für den Benutzer erschwert wird. Zudem muss auch die Symbolik nicht immer selbsterklärend sein, da gewisse Bedeutungen oft kulturabhängig different sind.

2.6.2 Pointing Devices

Als Pointing Device kann jede beliebige Hardware-Komponente eingesetzt werden, die es dem Benutzer ermöglicht, einfache bzw. kontinuierliche Dateneingaben am Computer durchzuführen. Shneiderman definiert in [53] verschiedene Interaktionsaufgaben, für die Zeigergeräte eingesetzt werden können:³⁶

- *Select*: der Benutzer wählt einzelne Objekte aus einem Pool aus.
- *Position*: der Benutzer wählt einen oder mehrere Punkte in einem mehrdimensionalen Raum. An Hand dieser Punkte können z.B. Objekte platziert werden.

³⁵Vgl. [53]: Shneiderman, 2005: S. 268-343

³⁶Vgl. [53]: Shneiderman, 2005: S. 358

- *Orient*: der Benutzer wählt eine Ausrichtung in einem mehrdimensionalen Raum durch Indikation einer Bewegungsrichtung.
- *Path*: der Benutzer führt eine Serie von Operationen durch, in der verschiedene Änderungen der Position sowie der Orientierung enthalten sind.
- *Text*: der Benutzer erzeugt, editiert und verschiebt Texteingaben. Durch das Zeigewerkzeug besitzt er die Möglichkeit, eindeutige Eingaben zu definieren.

Speziell in grafischen Benutzeroberflächen stellen Pointing Devices ein probates Mittel zur Interaktion und Manipulation dar. In vielen Quellen werden diese Handlungen dabei mit Formen der menschlichen Kommunikation verglichen und in erweitertem Zusammenhang als so genannte *Gestures* bezeichnet. Durch das Zeigen auf ein Objekt, lässt sich eine Verbindung zu diesem herstellen, um in weiterer Folge bestimmte Aktionen auf dieses wirken zu lassen. In der Welt der *HCI* spricht man in diesem Kontext von der Interaktionsform des *Point and Click*.³⁷ Interessante Objekte können mittels Pointing Device quasi ausgewählt und durch die Aktion des Klickens (vgl. Mouse als Pointing Device) zur weiteren Bearbeitung selektiert werden. Entscheidend ist hierbei, dass die Eigenschaften direkt manipuliert werden, sodass der augenblickliche Zusammenhang zwischen Aktion und Feedback für den Benutzer klar ersichtlich ist.



Abbildung 2.13: Beispiele für Pointing Devices an Hand von *Graphics Tablet* [33], *Mouse* [11] und *Digital Pen* [55].

Weitere Interaktionsformen, die sich an Hand dieser Aktionsabfolge klassifizieren lassen, sind z.B. das Selektieren und Verschieben (*Drag and Drop*), das Kopieren und Einfügen (*Copy and Paste*), sowie das Ausschneiden und Einfügen (*Cut and Paste*) von Objekten.³⁸ In [24] werden diese Handlungen

³⁷Vgl. [3]: Beigl Michael: *Point and Click – Interaction in Smart Environments*. University of Karlsruhe, 1999: S. 1

³⁸Siehe [24]: Herczeg, 2006: S. 112

von Herzeg als mentale Modelle bezeichnet, die der Benutzer aus seiner eigenen Erfahrungswelt heraus kennt. Er spricht in diesem Zusammenhang von den Grundbausteinen metaphorischer Interaktion.

Grundsätzlich bleibt zu sagen, dass sich sämtliche Interaktionsformen in Kombination mit Pointing Devices großer Beliebtheit bei den Benutzern erfreuen.

2.6.3 Tangible Interaction

Innerhalb eines *Tangible User Interfaces* wird dem Benutzer die Möglichkeit geboten, mit realen physischen Objekten in einer virtuellen Welt zu interagieren. Maßgebliche Herausforderung ist es hierbei, eine möglichst nahtlose Integration digitaler Daten in die reale Welt zu schaffen. Entscheidend ist, dass diese dabei in direkter Verbindung mit einer physischen Objektwelt stehen. Dadurch wird digitalen Daten eine gewisse Körperlichkeit verliehen, die der Benutzer quasi fühlen und auch manipulieren kann. Zusätzlich können den Objekten in der virtuellen Welt weitere Eigenschaften verliehen werden, welche durch visuelle und auditive Ergänzungen repräsentiert werden.

Zu den Pionieren auf diesem Gebiet zählt sich der Wissenschaftler Hiroshi Ishii – Professor am MIT (Massachusetts Institute of Technology) und Leiter der *Tangible Media Group*. Er spricht in [28] von so genannten *Tangible Bits*, die vom Benutzer mit Dingen des alltäglichen Lebens kombiniert und verbunden werden können. Das Ziel der *Tangible Bits* liegt darin, die Kluft zwischen Cyberspace und realer Umwelt zu überbrücken, indem man digitale Information greifbar macht.

Ishii führt hierzu Entwicklungsbeispiele an, die sich in drei Hauptbereiche gliedern:³⁹

- *Interactive Surfaces: Transformation of each surface within architectural space (e.g. walls, desktops, ceilings, doors, windows) into an active interface between the physical and virtual worlds;*
- *Coupling of Bits and Atoms: Seamless coupling of everyday graspable objects (e.g. cards, books, models) with the digital information that pertains to them; and*
- *Ambient Media: Use of ambient media such as sound, light, airflow, and water movement for background interfaces with cyberspace at the periphery of human perception.*

Um die gesamte Bandbreite der Interaktion mit den digitalen Informationen zugänglich zu machen, ist es notwendig, ihnen sämtliche Möglichkeiten zu bieten, diese mit Hilfe der physischen Objekte zu erfassen und direkt zu manipulieren. Durch entsprechende mediale Unterstützung können die Benutzer innerhalb des Handlungsablaufes geleitet und gefördert werden.

³⁹Siehe [28]: Ishii, 1997: S. 2



Abbildung 2.14: *Tangible Interaction* am Beispiel des ColorTable. Implementiert an der TU Wien. Aus [35].

2.6.4 Sound and Speech Interaction

Sound stellt im Kontext der Interaktion ein sehr komplexes Medium dar. Wird ein Ton, respektive Klang, in einem Raum über Lautsprecher wieder gegeben, so wird man – wenn auch nur indirekt und unterbewusst – davon beeinflusst. Anders als bei der Verwendung von visuellen Medien, die nur für Rezipienten mit direktem Sichtkontakt zur Quelle zugänglich sind, kann man sich der Wirkung nur sehr schwer entziehen. Abhilfe kann zwar durch entsprechende Kopfhörer bzw. Headsets geschaffen werden, allerdings eignen sich diese wiederum eher für Applikationen mit individuellem Charakter, da sie nicht unbedingt ein kooperatives Verhalten ermöglichen. Darüber hinaus muss man sich als Designer bewusst sein, dass Töne, Geräusche, Musik individuell wahrgenommen werden. Klangliche Elemente, die von manchen Benutzern als angenehm empfunden werden, können bei anderen dagegen Unbehagen hervorrufen. Anders als bei der visuellen Wahrnehmung ist es aber nicht so einfach, sich dem Sinneseindruck zu entziehen.

Die Verwendung von sprachlichen Elementen zum Zwecke der Interaktion wird in der Literatur oftmals als eine der besten, weil natürlichsten und benutzerfreundlichsten, Interaktionsformen bezeichnet. Speziell Systeme, die nur gelegentlich benutzt werden, stellen einen interessanten Anwendungsbereich dar. Durch die Verwendung der natürlichen Sprache bleibt es den Anwendern erspart, Interaktionswerkzeuge zu verwenden, mit denen potentiell fehlerbehaftete Eingaben erzeugt werden können. Herczeg führt in [24]

interessante Beispiele für die Vor- und Nachteile natürlicher Sprache als Interaktionsform an.⁴⁰

Vorteile natürlicher Sprachen:

- *natürliche Sprachen sind für unerfahrene Benutzer einfach anwendbar;*
- *natürliche Sprachen sind mächtig genug, um praktisch jede Information oder Anfrage zu kommunizieren;*
- *Benutzer haben keine Angst vor der Benutzung und fühlen sich dem Computer gegenüber überlegen;*
- *es muss keine spezielle formale Sprache für die Anwendung erlernt werden.*

Nachteile natürlicher Sprachen:

- *bis heute ist nur die Verarbeitung von Teilmengen natürlicher Sprache möglich;*
- *nur klar abgegrenzte, kleine Gegenstandsbereiche sind als Problembe-
reiche möglich;*
- *meist überschätzen die Benutzer die Leistungsfähigkeit der natürlich-
sprachigen Systeme;*
- *schwierige Sachverhalte lassen sich natürlichsprachig oft nur umständ-
lich ausdrücken;*
- *es ergeben sich Mehrdeutigkeiten auf der syntaktischen und semanti-
schen Ebene.*

Wie unschwer zu erkennen, ist die Konzeption und Realisierung natürlichsprachiger Systeme sehr aufwendig und hinsichtlich vieler Aspekte begrenzt. Im Gegensatz zur Interaktion mit geschriebener Sprache (vgl. Kommandosysteme, Abschnitt 2.2.4) wird die Aufgabe signifikant erschwert. Die Machbarkeit steht und fällt mit der Komplexität des Systems. Der verwendete Wortschatz muss stark eingeschränkt werden, was sich auf den Befehlsumfang einschränkend auswirkt. Zusätzlich ist es unerlässlich, genau zu prüfen in welchem Arbeitsumfeld diese Interaktionsform verwendet werden soll. Wie bereits erwähnt, kann man sich der auditiven Sinneswahrnehmung nur bedingt entziehen, sodass natürlichsprachige Kommunikation zwischen Mensch und Computer speziell für dritte Personen als störend empfunden werden kann. Darüber hinaus können auch Umgebungsgeräusche bzw. Lärm das Ergebnis negativ beeinflussen.

⁴⁰Siehe [24]: Herczeg, 2006: S. 107

2.6.5 Gesture Based Interaction

Gestiken zu verwenden, um mit einem Computer zu interagieren, stellt eine große Herausforderung dar. Zum einen sind zwar in den meisten Fällen keine speziellen Input-Devices zur Interaktion notwendig, zum anderen ist aber maßgeblich, dass menschliche Gestiken u.a. von Faktoren wie Persönlichkeit, physischen Fähigkeiten und kultureller Zugehörigkeit abhängig sind. In einem vorangegangenen Abschnitt wurde diese Thematik bereits kurz angesprochen. Beim Prinzip des *point and click* ist z.B. die Grundgestik des *Zeigens-auf-etwas* implementiert. Entscheidend für diesen Sachverhalt ist, dass diese Gestik in den meisten Kulturkreisen ähnlich interpretiert wird und damit in einem artgemäßen Kontext verwendet werden kann.

In weiterer Folge kann es zudem für gewisse Anwendungsfälle interessant sein, den menschlichen Körper als Interface einzusetzen. Die Vorteile liegen hierbei klar auf der Hand: die Benutzer sind schnell in der Lage, sich mit dem System auseinander zu setzen, da sie keine speziellen Vorrichtungen zur Dateneingabe benötigen, die ihrerseits unsachgemäß verwendet oder überhaupt beschädigt werden können. Darüber hinaus haben sie über ihren Körper die absolute Kontrolle.

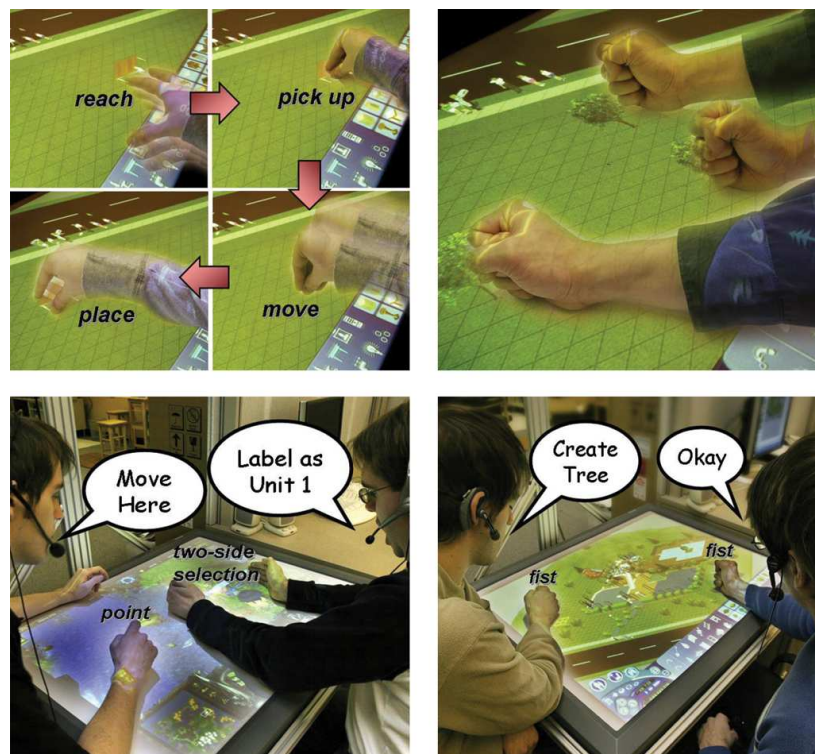


Abbildung 2.15: *Gesture Based Interaction* am Beispiel von *Multimodal Multiplayer Tabletop Gaming* von Edward Tse. Aus [58].

Zu beachten ist allerdings, dass es sehr schwierig ist, Gestiken bzw. Körperbewegungen und Positionen in einer herkömmlichen Umgebung zu detektieren. Die Mehrheit der momentan etablierten Techniken zur Gestik- und Bewegungserkennung ist von vielen äußeren Faktoren abhängig, die es zu kontrollieren gilt. Zusätzlich spielen aber auch erneut kulturelle Aspekte eine Rolle. Sehr viel Erfahrung und Wissen sind nötig, um in der Lage zu sein, die Körpersprache von Menschen zu analysieren und zu interpretieren. Zudem ist es auch notwendig zu erkennen, wie Menschen in speziellen Situationen reagieren und ihre Gefühle durch körperliche Reaktionen ausdrücken. Bezogen auf den Aspekt der Kulturalität muss man aber dennoch darauf hinweisen, dass keine „gemeinsame“ Kultur zwischen Mensch und Computer existiert, von der man eine generelle Körpersprache ableiten könnte. Intensive Forschungsarbeit wird auf diesem Gebiet betrieben, sowohl im Bereich der Sozialwissenschaften, als auch der technischen Wissenschaften. Um nun ein adäquates interaktives System zu kreieren, das sich der gestikbasierten Interaktion bedient, ist es essentiell, die Erkenntnisse aus beiden Bereichen einfließen zu lassen.

2.7 Zusammenfassung und Analyse der entscheidenden Erkenntnisse

Wie aus den vorangegangenen Kapiteln hervorgeht, sind die Formen und Varianten der Mensch-Computer-Kommunikation sehr vielfältig. Es wurden verschiedene Möglichkeiten geboten, um den Begriff des Interaktionsparadigmas in seiner Bedeutung einzugrenzen und damit greifbar zu machen. Entscheidend ist in diesem Zusammenhang, dass die präsentierten Modelle die Grundlage für die Konzeption und Implementierung interaktiver Systeme darstellen können. Sie öffnen quasi die Tür zu einem unendlichen Gestaltungsraum und zeigen gleichzeitig die Abgrenzungen auf, um den gewählten Weg durch diesen Raum sorgfältig und Ziel führend zu beschreiten. Da Interaktionsdesign ein weit reichender Begriff ist und viele verschiedene Wissensgebiete einbindet, war es unbedingt notwendig, gewisse Eingrenzungen vorzunehmen. Aus diesem Grunde wurde die gesamte Thematik ausschließlich aus Software- und medientechnischer Sicht betrachtet. Es stehen dem Designer in diesen Disziplinen verschiedene Möglichkeiten zur Verfügung, um den Benutzer in den Aktionsraum eines interaktiven Systems einzubinden. Die Konzepte und Modelle der Interaktion können dabei sehr vielfältig sein und die Auswahl muss mit Bedacht erfolgen, da dies eine bedeutende Entscheidung im gesamten Designprozess darstellt. Innerhalb dieser Arbeit wurde in diesem Zusammenhang der Begriff des Interaktionsparadigmas verwendet. Ziel war es, eine Definition zu finden und die charakteristischen Eigenschaften zu extrahieren. Dies stellte sich allerdings als schwierig heraus, da in der Literatur keine einheitliche Erklärung existiert. Um nun aber

eine Unterscheidung vornehmen zu können, wurden gewisse Grundkriterien definiert, die in der Einleitung zu diesem Kapitel ausführlich beschrieben wurden.

Wenn man nun die unterschiedlichen Beispiele, die vorangegangen präsentiert wurden, genau betrachtet, so ergeben sich zwei hauptsächliche Domänen, denen sich die angeführten Konzepte zuordnen lassen. Interaktionsparadigmen werden in der Literatur mehrheitlich verwendet, um System bzw. Kommunikationsmodelle zu beschreiben oder auch zu definieren. Grundsätzlich muss man hierbei sagen, dass eine eindeutige Differenzierung und Zuordnung allerdings nur sehr schwer vorgenommen werden kann, da viele Überschneidungen zwischen den Definitionsbereichen auftreten. Entscheidender Punkt ist, dass das Interaktionsparadigma vom entsprechenden Anwendungskontext abhängig ist. Dies betrifft sowohl die Auswahl des Konzeptes, als auch die Zuordnung.

Die Theorien und Konzepte von Michael Herzog, der ja zuallererst angeführt worden ist, sind z.B. vom Modell des Computers als Kommunikationspartner geprägt. Mensch und Computer sind demnach in der Lage, wie Menschen untereinander zu kommunizieren. Ein entstehender Dialog ist dabei an einen abgegrenzten Handlungsraum gebunden, in dem sich der Benutzer frei bewegt und Aktionen ausführt. Gleichzeitig liefert Herzog aber auch Definitionen, die sich dem Konzept der systembezogenen Modelle zuordnen lassen, indem er den Begriff der Systemparadigmen prägt, die wiederum unterschiedliche Varianten der Mensch-Computer-Kommunikation beschreiben.

Jenny Preece schlägt mit ihren Definitionen von Interaktionsparadigmen in eine ähnliche Kerbe. Ihre Ansichten stellen dabei eine interessante Alternative zu den anderwärtigen Konzepten dar. Dies betrifft in erster Linie die Praxis ihrer Arbeit, die intensiv, gemeinsam mit den Benutzern durchgeführt wird und sich sehr stark an deren Bedürfnissen orientiert. In diesem Zusammenhang spricht Preece von konzeptionellen Modellen und spezifischen Interfacemetaphern. Ziel ist es dabei, Kenntnisse der Anwender aus dem wirklichen Leben auf die Kommunikation mit einem Computer-gestützten System umzulegen. Sie definiert das Interaktionsparadigma als technisches Konzept, das im Rahmen des Interaktionsdesigns umgesetzt werden kann. Speziell die neuen Technologien bieten den Designern multiple Möglichkeiten, um ihre Ideen in die Tat umzusetzen.

Shneiderman bzw. Raskin sind mit ihren Konzepten zwar ebenfalls sehr stark am Systemmodell orientiert, allerdings lassen sie sich von der Arbeitspraxis her, mehrheitlich dem Bereich der Softwareergonomie zuordnen. Vor allem Shneiderman ist durch seine Arbeit mit Theorien und Modellen in der Lage, die komplexen Abläufe im Verhältnis zwischen Mensch und Computer zu deuten und greifbar zu machen. Er befasst sich sehr intensiv mit den Varianten des User-Interface-Designs, sowie der systemgestützten Benutzerführung und verweist in diesem Kontext auf so genannte Interakti-

onsstile, die die primären Basismodelle der *HCI* repräsentieren. Wenn man nun die im entsprechenden Kapitel angeführten Konzepte näher betrachtet, so erkennt man, dass diese doch eher Software-orientiert sind. Im Gegensatz zu Preece, deren Ansichten mehrheitlich auf dem technischen Konzept im Hintergrund beruhen, hakt Shneiderman eine Stufe darüber ein, indem er die Fragen der technischen Umsetzung außen vor lässt und sich rein auf die ergonomischen Aspekte von Software konzentriert.

Jef Raskin ist mit seinen Konzepten – wie bereits erwähnt – ähnlich einzuordnen. Er vertritt grundsätzlich die Meinung, dass die kompromisslose Auseinandersetzung mit der digitalen Technologie vorbei ist. Es besteht keine Bereitschaft bei den Benutzern, für das Durchführen einfachster Aufgaben eine Vielzahl von komplexen Arbeitsschritten zu erlernen. Raskin versucht bei der Lösung dieses Sachverhaltes seine Erfahrungen aus der sozialwissenschaftlichen Forschung einfließen zu lassen. Das Ziel sollte es sein, Interfacemethodologien zu kreieren, die eine geringe Anzahl von Interaktionsmöglichkeiten enthalten, die sich aber an eine Vielzahl von unterschiedlichen Anwendungsbereichen anpassen lassen. Er definiert in diesem Zusammenhang das Interfacekonzept der *Zooming Interfaces*. Entscheidender Aspekt hierbei ist, dass auf die starren Verzeichnis- und Dateistrukturen gänzlich verzichtet wird und die Informationen, in Datenblöcken organisiert, jederzeit frei zugänglich sind. Durch die Tätigkeit des *Zooming* können einzelne Teilbereiche vergrößert bzw. verkleinert werden, wodurch es möglich wird, die gesamte Informationsebene zu erkunden. Interessant an Raskin's Konzepten und Modellen ist, dass er seine langjährige wissenschaftliche und praktische Erfahrung erkennbar mit einbringt. Er übt Kritik, liefert aber auch Ideen und Alternativen.

Interaktionsparadigmen sind also in gewisser Weise als Modelle von Kommunikation bzw. Systemen zu betrachten. Wie nun aber die direkte Interaktion innerhalb dieser Modelle erfolgen kann, war – die Konzepte von Shneiderman ausgenommen – eher weniger Bestandteil der Betrachtung. Da die unterschiedlichen Interaktionsformen ein wesentlicher Bestandteil im Entwicklungsprozess eines interaktiven Systems sind, ist es unbedingt notwendig auf Aspekte der Multimodalität verstärkt einzugehen und darzustellen, wie nun die direkte Interaktion zwischen Mensch und Computer realisiert werden kann. Die in Abschnitt 2.6 angeführten Beispiele sind dabei nur als Auszüge der Möglichkeiten zu betrachten, die den Designern und Entwicklern potentiell zur Verfügung stehen und stellen in weiterer Folge nur eine Ergänzung der bereits besprochenen Thematik dar.

Zusammenfassend bleibt nun zu sagen, dass der Begriff des Interaktionsparadigmas nur schwer greifbar ist, da er in der Literatur zur Definition verschiedener Sachverhalte herangezogen wird. Inhaltlich treten aber dennoch starke Kongruenzen auf, wodurch sich eine grundlegende Unterscheidung in kommunikationsorientierte bzw. systemorientierte Konzepte, in Kombina-

tion mit unterschiedlichen Interaktionsformen vornehmen lässt. Ein Beispiel hierfür ist das Prinzip der direkt manipulativen Systeme in Kombination mit Pointing Devices zur Interaktion.

Unter dem Aspekt, dass es auch ein erklärtes Ziel dieser Arbeit ist, ein Interaktionsparadigma zu finden, welches sich auch im Rahmen einer praktischen Implementierung umsetzen lässt, stellt sich die Frage was wirklich passend ist. Hierbei ist klar ersichtlich, dass sich vorrangig die Modelle und Ideen von Shneiderman bzw. Raskin sehr gut eignen, da diese eher anwendungsorientiert aufgebaut sind. Bei Shneiderman bleibt zu sagen, dass er sich mit durchaus etablierten und oftmals implementierten Konzepten auseinandersetzt. Nicht zuletzt haben z.B. Menüauswahl und Formularfelder speziell im Bereich des WWW eine Renaissance erlebt.

Raskin hingegen hat auf Basis seiner sozialwissenschaftlichen Erfahrungen Theorien und Modelle kreiert, die durchaus interessante Ansätze beinhalten. Vor allem sein Konzept der *Zooming Interfaces* ist ein Designansatz, der eine interessante und wertvolle Basis für eine Implementierung darstellt. Aus diesem Grund wird von sämtlichen in dieser Arbeit angeführten Beispielen, das Interfacekonzept der *Zooming Interfaces* für eine praktische Umsetzung ausgewählt.

Der gesamte Arbeitsprozess, von der Erstellung des Grunddesigns bis hin zum finalen Prototypen, findet sich in den nachfolgenden Kapiteln genau dokumentiert.

Kapitel 3

Designkonzept

Ein primäres Ziel dieser Arbeit war es, die Ergebnisse aus dem Forschungsteil in Form einer Prototypenimplementierung in die Praxis umzusetzen. Ausgehend von den Erkenntnissen aus den vorangegangenen Kapiteln, bieten sich hierzu viele verschiedene Möglichkeiten. Das Für und Wider der einzelnen Konzepte wurde schon genau erläutert.

Grundsätzlich fiel die Wahl auf die Modelle und Theorien von Jef Raskin. Das von ihm beschriebene Prinzip der *Zooming Interfaces* stellt eine passende Basis für eine interessante und spannende Implementierung dar.

Auf den folgenden Seiten werden sämtliche Teilbereiche der Aufgabendurchführung genauer vorgestellt und analysiert. Von der grundsätzlichen Formulierung der Designideen über die Skizzierung verschiedener Interaktionsmethoden bis hin zur finalen Implementierung. Ein Hauptaugenmerk liegt dabei vor allem darauf, die Vorgehensweise innerhalb der praktischen Umsetzung von Raskin's theoretischen Modellen strukturiert zu veranschaulichen. Es werden hier auch Schwierigkeiten und Hindernisse dokumentiert und nicht zuletzt die persönlichen Erfahrungen.

Ziel dieses Abschnittes ist es, eine mögliche Umsetzungsvariante des Interfacekonzeptes der *Zooming Interfaces* zu präsentieren. Dadurch soll die Möglichkeit geboten werden, Stärken und Schwächen genau zu erkennen, um in weiterer Folge einen Ausblick geben zu können, wie sich dieses Prinzip weiter entwickeln und an Bedeutung gewinnen kann.

3.1 Ausgangsposition

Der Begriff des Interaktionsparadigmas wird mehrheitlich verwendet, um Kommunikations- bzw. Systemmodelle zu beschreiben. Dies ist eine der Erkenntnisse aus den vorangegangenen Abschnitten. Wie die tatsächliche Interaktion bzw. Kommunikation zwischen Mensch und Computer an sich erfolgt, ist oftmals weniger Bestandteil der Betrachtung. Ähnlich verhält es sich auch beim Konzept der *Zooming Interfaces*. Raskin bietet in sei-

nen Ausführungen genauen Einblick in das Grundprinzip der Interaktion im *Zoom Space*, das mit der Tätigkeit des *Zooming* beschrieben wird. Durch die Aktion des *Zoom In* wird es möglich, einzelne Bereiche im Detail zu betrachten, um in weiterer Folge durch die Aktion des *Zoom Out* wiederum einen klaren Überblick über das Gesamtsystem zu erlangen. Er bietet allerdings berechtigterweise keine detaillierten Informationen darüber, wie der Benutzer einzelne Handlungen setzen kann. Kurz gesagt: die grundsätzliche Interaktionsform ist auf Basis seiner Betrachtungsweise vorerst irrelevant. Im Rahmen der praktischen Implementierung stellte sich diese Frage aber sehr wohl. Grundsätzlich sind in Abschnitt 2.6 Formen der Multimodalität beschrieben, die auch für die Interaktion im *Zooming Interface* durchaus adäquat sein können. Es galt abzuwägen welches Prinzip sich am besten eignet. Sprache oder Sounds als Eingabemedium zu verwenden, hätte in Bezug auf Verarbeitung von Input und Output ein gewisses Maß an Komplexität verursacht. Des Weiteren wären an Benutzer und System sehr hohe Anforderungen gestellt worden. Die Interaktion mittels Pointing Devices war dabei schon eher relevant, da diese für eine Vielzahl von Aufgaben eingesetzt werden können (siehe Abschnitt 2.6.2). Allerdings tritt diese Interaktionsform primär in Kombination mit herkömmlichen Desktop-basierten Systemen auf. Da Raskin das *ZIP* aber als Erweiterung zu diesem Modell sieht, erschien es als unumgänglich von diesem konventionellen Prinzip der Interaktion abzuweichen. Die Konzepte der *Tangible User Interfaces* bzw. der *Gesture Based Interaction* erschienen – im Sinne einer Implementierung mit innovativem Charakter – als Ziel führend. Aus diesem Grunde wurde das Interfacekonzept der *Zooming Interfaces* sowohl in Kombination mit *Tangible Interaction* als auch *Gesture Based Interaction* praktisch umgesetzt, um in weiterer Folge einen Vergleich zwischen beiden Applikationen ziehen zu können.

Raskin setzt sich in seinen Publikationen intensiv mit gängigen Praktiken der Benutzerinteraktion auseinander. Er bezieht sich in diesem Zusammenhang vor allem auf das Prinzip der Desktop-basierten Systeme, die seiner Meinung nach den Anwender einschränken und dadurch in weiterer Folge am Ende ihrer Entwicklung angelangt sind. Das *ZIP* sieht er dabei nicht im kompletten Gegensatz, sondern mehrheitlich als Erweiterung. Die Interaktion am Desktop wird damit auf eine neue Ebene transferiert. Demzufolge erschien es bei der Umsetzung als notwendig, das Prinzip der Desktop-Metapher auch in die Implementierung zu integrieren. In welcher Form diese definierte Vorgabe umgesetzt wurde zeigt sich im folgenden Kapitel.

3.2 Konzeptionierung des interaktiven Systems

Ein wichtiger Aspekt im Rahmen der praktischen Umsetzung war es, eine Interaktionsmöglichkeit zu schaffen, die vielseitig einsetzbar ist. Der menschliche Körper sollte dabei quasi als Input Device fungieren, wodurch die Möglichkeit geboten werden sollte, mittels Arm- und Handbewegungen bzw. Gestiken zu interagieren. Das Erkennen dieser Bewegungen – um die Kommunikation mit einem Computer zu ermöglichen – stellt grundsätzlich eine große Herausforderung dar (siehe Abschnitt 2.6.5). Auf Basis der Erfahrungen aus vorangegangenen Projekten in ähnlichen Bereichen, wurde ein kamerabasiertes Erkennungssystem (Englisch: *Tracking System*) konstruiert. Grundsätzlich werden dabei die Körperbewegungen von einer Kamera aufgenommen und in weiterer Folge innerhalb einer Softwareapplikation analysiert, mit dem Zweck intendierte Handlungen des Benutzers zu extrahieren. Im Bereich des Interaktionsdesigns wird die analytische Studie von menschlichen Bewegungen ein immer wichtigerer Bestandteil der Forschung. Das Erkennen von menschlichen Personen und deren Position, das Messen und Interpretieren von deren Haltungen und Bewegungen, werden dabei zu zentralen Aufgabenstellungen. Zusätzlich ist es natürlich auch notwendig, die erhaltenen Daten sinnvoll weiterzuverarbeiten. Die Arbeitstechniken, die dabei zur kamerabasierten Bewegungserkennung eingesetzt werden, sind teilweise sehr unterschiedlich. Ein etablierter Ansatz ist z.B. das so genannte *Motion Sensing*¹, ein Verfahren bei dem nach Veränderungen in den von der Kamera generierten Einzelbildern (Englisch: *Frame*) gesucht wird. Um diese Bewegungen verfolgen zu können (Englisch: *tracken*) werden Differenzbilder von aufeinander folgenden Frames erzeugt, die auf Veränderungen im Muster des Bildes hin untersucht werden. Ein weiteres etabliertes Verfahren der Bewegungserkennung ist das so genannte *Object Tracking*.² Hierbei wird die von der Kamera erzeugte Bildfolge auf Farben bzw. Formen hin untersucht, die in Verbindung mit bestimmten Objekten stehen.

Grundsätzlich ist aber zu erkennen, dass die meisten Verfahren mit der Verwendung von so genannten Markern operieren. Dies bedeutet, dass interessante Objekte bzw. Extremitäten mit entsprechenden markanten Indikatoren versehen werden. Je nach Prinzip, das zur Analyse der gewonnenen Bilddaten herangezogen bzw. implementiert wird, besitzen diese Marker eine unterschiedliche Beschaffenheit. Sie müssen sich z.B. in Form oder Farbe signifikant vom restlichen Bild unterscheiden. Dadurch kann gewährleistet werden, dass deren Positionen genau geortet und weiterverarbeitet werden können.

Im Fall, der in dieser Arbeit beschrieben wird, erfolgt die eigentliche Bewegungserkennung durch den Einsatz spezieller Farbmarker, die entweder

¹Vgl. [19]: Eisenstein Jacob, Mackay Wendy: *Interacting with Communication Appliances*. MIT, 2006: S. 2

²Vgl. [19]: Eisenstein, Mackay, 2006: S. 2

an den Händen in Form von Handschuhen getragen werden (*Gesture Based Interaction*) oder an entsprechenden Objekten angebracht sind, die zur Interaktion verwendet werden können (*Tangible Interaction*). Das Kamerabild wird auf diese Farben hin untersucht, um so genaue Positionsangaben bezüglich der Marker zu erhalten. Da dieses Prinzip des Object-Trackings die gestellten Anforderungen an Geschwindigkeit und Genauigkeit der Datenverarbeitung erfüllte, wurde es für die Implementierung der Benutzerinteraktion herangezogen. Zusätzlich war auch der damit verbundene geringe finanzielle Aufwand mit entscheidend, da die Hauptinvestition mit dem Ankauf einer handelsüblichen WebCam abgetan war.

Vorab ist zu sagen, dass das eingesetzte Verfahren auch für gewisse Einschränkungen und Anfälligkeiten bekannt ist. Bezogen auf die Bewegungserkennung über Kamera ist vor allem entscheidend, dass die Resultate mit der Qualität des verwendeten Gerätes einhergehen. Je höher diese ist, desto besser werden die gewonnenen Daten sein und desto besser können diese innerhalb der Applikation zu einem gewünschten Ergebnis weiter verarbeitet werden. Ein weiteres Manko ist die Abhängigkeit von äußeren Faktoren. Um ein optimales Bild zu gewährleisten, ist es notwendig, dass der Hintergrund möglichst kontrastarm ist und der Raum entsprechend hell ausgeleuchtet wird. Zusätzlich spielt die Auswahl der Farbmarker eine bedeutende Rolle, da sich diese markant vom restlichen Bild abheben müssen, um eine genaue Detektion der Bewegung gewährleisten zu können.

3.2.1 Setup

Um nun ein reibungsloses Tracking zu garantieren, war es notwendig, ein Setup zu kreieren, das die Abhängigkeiten von den äußeren Faktoren möglichst gering hält. Als Vorbild diente das System des *ColorTable*. Es handelt sich dabei um ein *Tangible User Interface*, das im Rahmen des IST-2006 IPCity-Projektes am Institut für Gestaltungs- und Wirkungsforschung der Technischen Universität Wien implementiert worden ist.³ Das System besteht grundsätzlich aus einem drehbaren Tisch, auf dem verschiedene Marker beliebig angeordnet werden können. Darüber befindet sich in entsprechendem Abstand eine Kamera, die in gewisser zeitlicher Frequenz Abbildungen der gesamten Szenerie erzeugt. Auf dieser Basis können nun beliebige Softwareapplikationen kreiert werden. Entscheidend ist dabei, dass die physischen, realen Marker in direkter Verbindung mit einer virtuellen Objektwelt stehen (siehe Abschnitt 2.6.3). Für den Benutzer kann der gesamte Interaktionsprozess durch entsprechende auditive (z.B. Lautsprecher) oder visuelle (z.B. Projektion) Repräsentationen demonstriert werden.

Für die praktische Umsetzung im Rahmen dieser Arbeit wurde ein modifiziertes Setup implementiert, wobei aus dem System des ColorTables wichtige

³Vgl. [35]: Maquil Valerie: *Tangible Interaction in Mixed Reality Applications*. Vienna University of Technology, 2006

Anhaltspunkte gewonnen werden konnten. Primär war dabei das Konzept des Tisches als interaktive Umgebung inspirierend. Raskin spricht auch in seinen Ausführungen zum *ZIP* vom Prinzip der Desktop-basierten Systeme, dem in gewisser Weise eine Erweiterung widerfährt, speziell was das Potential der Interaktion betrifft. Durch die Implementierung des interaktiven Systems in Form eines Tisches, war damit auch eine interessante Überleitung zur Desktop-Metapher gegeben.



Abbildung 3.1: Interaktives System – Gesamtsetup

Zur Konzeptionierung lässt sich sagen, dass es sich beim Gesamtsystem um einen Prototypenaufbau handelt. Ein wichtiges Kriterium war, die aufkommenden Materialkosten gering zu halten, aber gleichzeitig eine bestmögliche Umsetzung der Desktop-Metapher zu erreichen. Aus diesem Grund wurde für die Fixierung der Kamera, die ein zentrales Element im beschriebenen Tracking-System darstellt, eine Montagevariante in Kombination mit einer Schreibtischlampe gewählt. Zum einen war damit ein Utensil in Verwendung, das zum Repertoire einer herkömmlichen Büroarbeitsumgebung zählt. Zum anderen konnte damit auch eine verbesserte Ausleuchtung der Tischfläche erreicht werden, was sich positiv auf den Prozess des Trackings auswirkt. Die visuelle Repräsentation der Interaktion erfolgt für den Benutzer in Form einer Projektion, die auf eine Wand abgestrahlt wird.

Zu Beginn der praktischen Umsetzung bestand auch die Möglichkeit, die Projektion direkt über dem Tisch – mit diesem als reflektierende Fläche – durchzuführen. Allerdings hätte sich damit einerseits der Montageaufwand beträchtlich erhöht, andererseits wäre auch der Prozess des Object-Trackings erschwert worden. Zum aktuellen Zeitpunkt passiert dies durch eine Analyse der Kamerabilder mit der einfarbigen Tischfläche im Hintergrund (im Prototypenaufbau: weiß), wodurch die Marker von der Software sehr gut erkannt und die Positionen exakt ermittelt werden können. Dies hat eine höhere Genauigkeit der Benutzereingaben – durch Bewegung der Hände bzw. Verschiebung der Marker – zur Folge. Würde die Projektion der Inhalte direkt auf den Tisch erfolgen, wäre die Extraktion der Markerpositionen aufgrund möglicher Farbgleichheiten ungleich schwieriger.

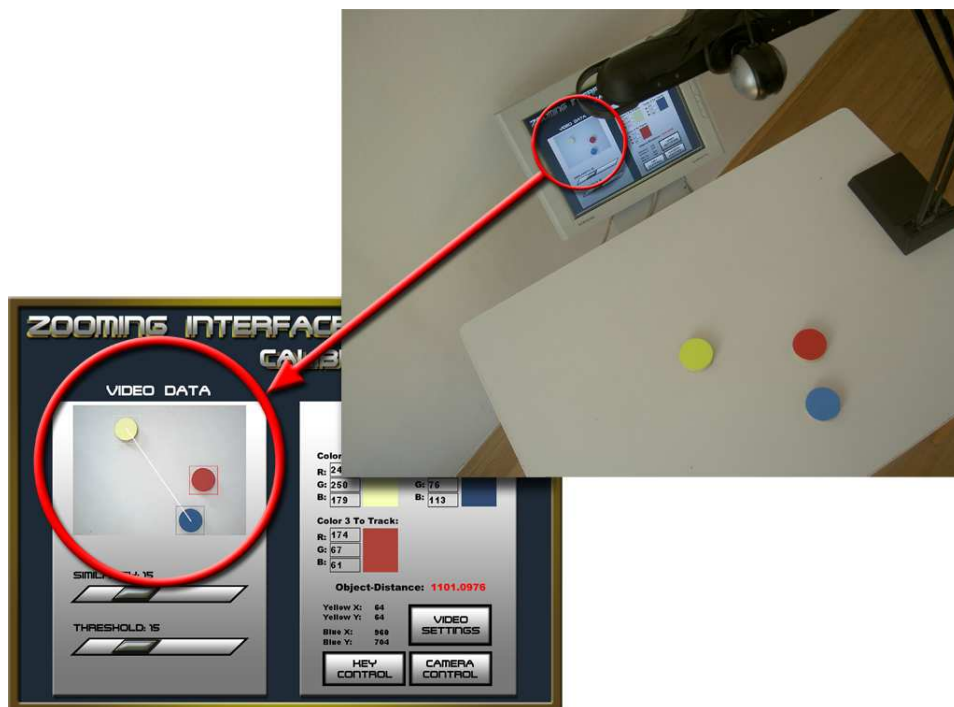


Abbildung 3.2: Interaktives System – Object-Tracking im Modus *Tangible Interaction*.

Ein weiteres wichtiges Element des Setups stellt der Informationsmonitor dar, der an einer der Seitenkanten des Tisches, in Blickrichtung des Benutzers, montiert ist. Darauf werden sämtliche Daten, die für das Tracking relevant sind, angezeigt. Es wird hier z.B. auch das Kamerabild eingeblendet, um den Benutzer über die aktuellen Positionen der Marker zu informieren. Darüber hinaus bietet sich die Möglichkeit, durch entsprechende Konfigurationen Software-basierte Feinabstimmungen am Trackingsystem vorzunehm-

men. Genauere Informationen zum Design des Info-Bildschirmes finden sich in Abschnitt 4.3.



Abbildung 3.3: Interaktives System – Informationsmonitor

3.2.2 Object-Tracking – Farbmarker

Wie bereits erwähnt, ist das Verwenden von Markern ein zentrales Element im Konzept des Object-Trackings. Wichtigste Anforderung an diese Marker ist die Möglichkeit einer klaren Unterscheidung. Im beschriebenen Fall erfolgt diese Differenzierung auf Basis unterschiedlicher Farben, die sich markant von den restlichen Elementen des Kamerabildes abheben müssen, um eine genaue Detektion der Positionsverschiebungen gewährleisten zu können. Die Form der Marker spielt in diesem Zusammenhang eine untergeordnete Rolle.

Im Vorfeld der Implementierung war klar, dass es bei einer angemessenen praktischen Umsetzung des *ZIP* notwendig sein wird, mehrere Punkte – und damit verbunden mehrere Farben – gleichzeitig zu tracken. Im Fall der *Gesture Based Interaction* war es eine grundsätzliche Anforderung, die Interaktion mit den Händen zu ermöglichen. Durch das *Zeigen auf ein Objekt* lässt sich eine Verbindung zu diesem herstellen (siehe Abschnitt 2.6.2 Pointing Devices, 2.6.5 Gesture Based Interaction). Als Farbmarker fungieren daher herkömmliche Handschuhe, die vom Benutzer bequem an den Händen getragen werden können.



Abbildung 3.4: Object Tracking – Farbmarker *Gesture Based Interaction*

Im Fall der *Tangible Interaction* sind die Farbmarker in Form von speziellen Objekten ausgeführt, die zur Interaktion herangezogen werden. Diese sollten einfach zu greifen und dadurch praktikabel handhabbar sein. Da innerhalb der Softwareapplikation keine gezielte Unterscheidung in Bezug auf die Form notwendig ist, sind sämtliche Marker kreisrund gefertigt.



Abbildung 3.5: Object Tracking – Farbmarker *Tangible Interaction*

3.3 Entwurf und Skizzierung

Im Vorfeld der Implementierung stellte sich die Frage, in welchen Bereichen sich das *ZIP* in adäquater Weise anwenden ließe. Raskin beschreibt in [46] exemplarisch die Umsetzung in Kombination mit einer Web-Applikation. Die Informationen sind dabei auf einer endlichen Fläche – dem *Zoom Space* – in verschiedenen Datenblöcken organisiert. Durch die Interaktionstechniken des *Zoom In* und *Zoom Out* ist es möglich, diese Datenböcke zu erkunden und entsprechend zu modifizieren (siehe Abschnitt 2.4.3). Bezüglich weiterer möglicher Einsatzbereiche hält sich Raskin in seinen Ausführungen eher bedeckt. Interessant ist im Zusammenhang mit dem *ZIP* aber das Prinzip der universalen Interaktionsoberfläche, auf der einschränkende Elemente auf ein Minimum reduziert und Verzeichnisstrukturen vollständig eliminiert sind. Legt man dieses Prinzip auf ein gängiges Desktop-basiertes System mit grafischer Benutzeroberfläche um, würde man einen quasi unendlichen Desktop vorfinden.

Dieses Konzept stellte auch den Anstoß für den konkreten Anwendungsfall dar, der im Rahmen der praktischen Umsetzung des *ZIP* implementiert wurde. Es entstand die Idee von einer grafischen Ebene (in weiterer Folge auch als Arbeitsbereich bzw. -fläche bezeichnet), auf der verschiedenen Dateiobjekten (z.B. Audio, Video, Bilder) beliebig positioniert bzw. organisiert sind. Auf dieser Fläche kann frei navigiert werden, um sämtliche Bereiche zu erkunden. In Kombination mit den Interaktionsformen des *Zoom In* und *Zoom Out* wird eine beliebige Vergrößerung bzw. Verkleinerung erreicht. Ein wichtiger Aspekt ist hierbei, dass sich nach Auswahl eines Dateiobjektes das Interface automatisch dem Dateitypen anpasst und adäquate Möglichkeiten der Bearbeitung anbietet. Sämtliche Elemente, die den Benutzer in seiner Arbeitstätigkeit einschränken können, sollen damit auf ein Minimum reduziert werden. Es soll keine Zeit vergeudet werden mit dem Suchen nach Verzeichnissen oder Dateien und den zugehörigen Applikationen. Die volle Konzentration gilt dem Lösen der gestellten Aufgabe.

Für die praktische Umsetzung im Rahmen dieser Arbeit galt die Implementierung des Navigations- und Selektierungsinterfaces, in Kombination mit den Interaktionstechniken *Zoom In* und *Zoom Out*, als Grundanforderung. In den folgenden Abschnitten finden sich einige Skizzen und grafische Abbildungen, die im Zuge des Entwurfsprozesses angefertigt wurden. Diese dienen primär jenem Zweck, einen ersten Eindruck des Grunddesigns zu liefern und vorab schon eine Definition möglicher Problemfelder zu geben.

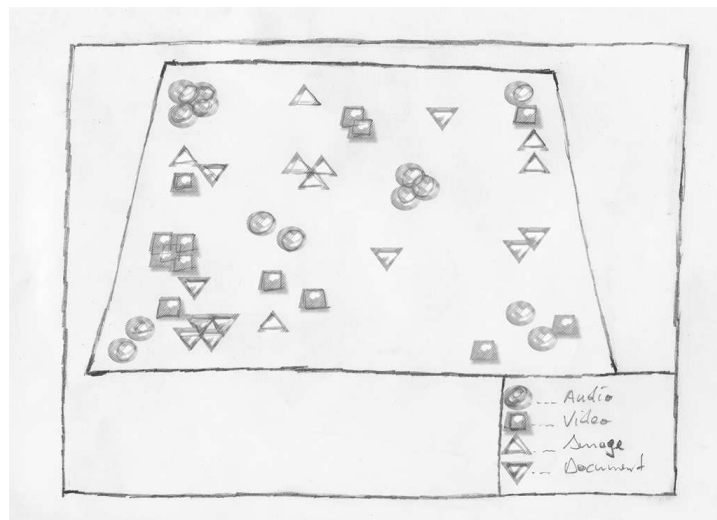


Abbildung 3.6: Erster Entwurf der grafischen Oberfläche mit einer Visualisierung der Dateiobjekte durch entsprechende geometrische Formen.

3.3.1 Gesture Based Interaction

Als zentrale Interaktionstechnik im *ZIP* gilt das Prinzip des *Zooming*. Im Rahmen des Designprozesses wurde festgelegt, hierfür den Abstand zwischen den Händen heran zu ziehen. Bewegt man diese auseinander entspricht dies der Tätigkeit des *Zoom In*. Bewegt man sie wiederum zusammen so kommt es zum *Zoom Out*.

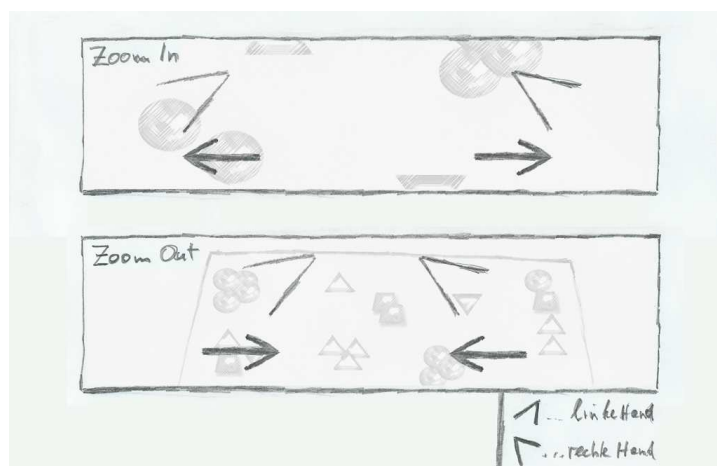


Abbildung 3.7: *Gesture Based Interaction* – Grafischer Entwurf für die Aktion des *Zooming*.

Da sich im Fall des *Zoom In* der Arbeitsbereich vergrößert, muss auch die Möglichkeit geboten werden zu navigieren. Die Navigation erfolgt in vier Richtungen: nach oben, unten, rechts und links. Zu diesem Zweck werden im Zuge der Implementierung der grafischen Oberfläche an den Aussenkanten aktive Bereiche festgelegt, in denen die Hände positioniert werden müssen, um eine entsprechende Verschiebung des Arbeitsbereiches in die gewünschte Richtung zu erreichen.

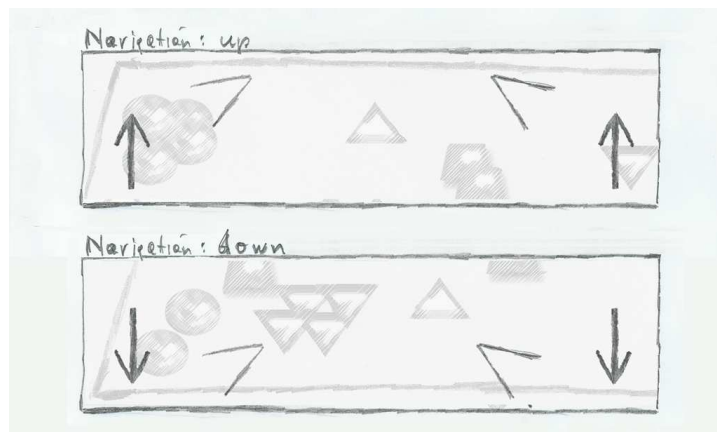


Abbildung 3.8: *Gesture Based Interaction* – Grafischer Entwurf für die vertikale Navigation.

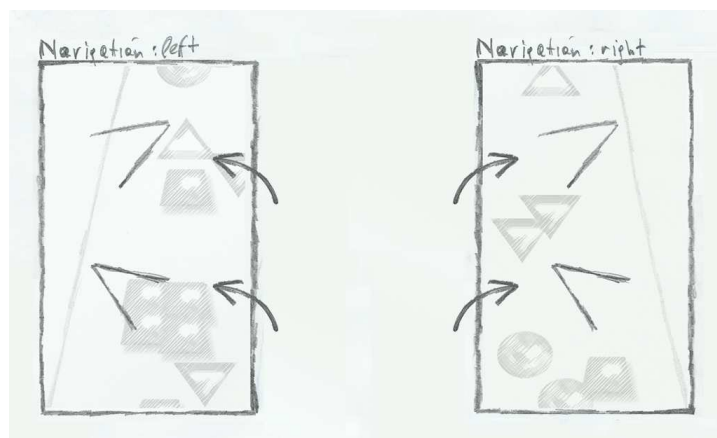


Abbildung 3.9: *Gesture Based Interaction* – Grafischer Entwurf für die horizontale Navigation.

Zusätzlich zu diesen elementaren Benutzeraktionen, die notwendig sind, um eine ordnungsgemäße Interaktion zu garantieren, ist es auch möglich die grafische Oberfläche beliebig zu rotieren. Dadurch soll dem Anwender die größtmögliche Bewegungsfreiheit gegeben werden. Im Sinne eines möglichen zukünftigen Einsatzes in kollaborativen Arbeitssituationen, wäre dies ein interessanter Ansatz, da jeder Partizipierende die Arbeitsfläche entsprechend seiner Blickrichtung in Position bringen kann. Im Designprozess wurde festgelegt, für die Rotation jenen Winkel heranzuziehen, der sich durch das Drehen bzw. Verschieben der Hände ergibt.

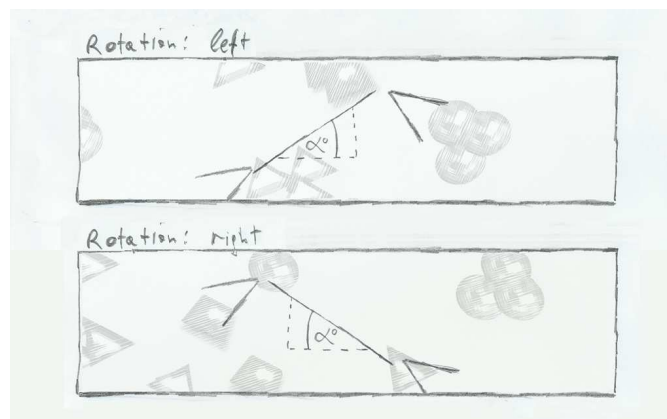


Abbildung 3.10: *Gesture Based Interaction* – Grafischer Entwurf für die Rotation des Arbeitsbereiches.

Um nun den Ablauf der Interaktion zu komplettieren, war die Festlegung des Selektierungsmodus ausständig. Dies wurde folgendermaßen gelöst: um eine Auswahl innerhalb der Arbeitsfläche durchführen zu können, ist es notwendig eine Hand über dem ausgesuchten Element und die zweite in einem gewissen Mindestabstand zu platzieren.

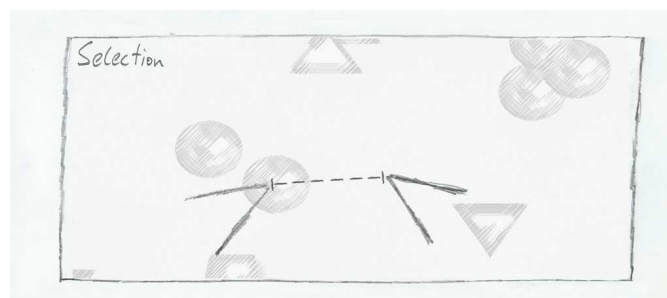


Abbildung 3.11: *Gesture Based Interaction* – Grafischer Entwurf für die Auswahl eines Elements.

Grundsätzlich bestanden gewisse Zweifel, ob diese Form der Auswahl problemlos funktionieren würde. Anforderung war, dass die Selektierung eindeutig und für den Benutzer unmissverständlich ist. Klarheit über die ordnungsgemäße Funktion des hier beschriebenen Prinzips sollte die praktische Umsetzung liefern.

Im Designprozess wurde versucht einen angemessenen und vor allem eindeutigen Weg der Interaktion zu finden, der sich in Kombination mit dem in Abschnitt 3.2.1 beschriebenen interaktiven System anwenden lässt. Ob sich die in Form von Skizzen aufgezeichneten Ansätze nun wirklich bewähren würden, sollte sich anschließend an die Implementierung in einer entsprechenden Testphase zeigen.

3.3.2 Tangible Interaction

Bevor zum Kapitel der praktischen Implementierung übergegangen wird, ist es notwendig auch den Gestaltungsansatz in Bezug auf *Tangible Interaction* darzulegen. Im Unterschied zur *Gesture Based Interaction* werden die Positionen spezieller physischer Objekte zur Interaktion herangezogen. Dies hat zur Folge, dass sämtliche Aktionen des Benutzers durch eine Manipulation dieser Objekte zu erfolgen hat.

Im Anfangsstadium stellte sich in diesem Zusammenhang die Frage, wie viele Objekte dem Anwender zur Verfügung gestellt werden sollen bzw. für eine passende Interaktion notwendig sind. Grundsätzlich war klar, dass *Zooming* und Navigation wieder auf herkömmlicher Basis mit zwei Objekten durchgeführt werden können. Entscheidend ist, dass diese in direkter Verbindung mit ihren virtuellen Pendants stehen und sich eine Manipulation unmittelbar und direkt auswirkt. Es wurde ein Ansatz gewählt, bei dem beide Objekte den Ausschnitt der Ansicht auf die Arbeitsfläche abgrenzen. Im Initialzustand sind sie an den äußeren Eckpunkten – diagonal gegenüberliegend – positioniert. Durch eine Veränderung der Distanz zwischen den beiden kann der Ausschnitt entsprechend vergrößert bzw. verkleinert werden.

Eine horizontale bzw. vertikale Verschiebung der Objekte hat eine entsprechende Translation der grafischen Oberfläche zur Folge. Dadurch wird dem Benutzer die Möglichkeit zur Navigation geboten, wodurch sämtliche Abschnitte des Arbeitsbereiches erkundet werden können. Wichtig ist, dass zu diesem Zweck beide Objekte gleichzeitig in die selbe Richtung verschoben werden müssen, um eine Abstandsänderung zwischen diesen zu unterbinden. Dies hätte ansonsten eine Skalierung der grafischen Oberfläche zur Folge.

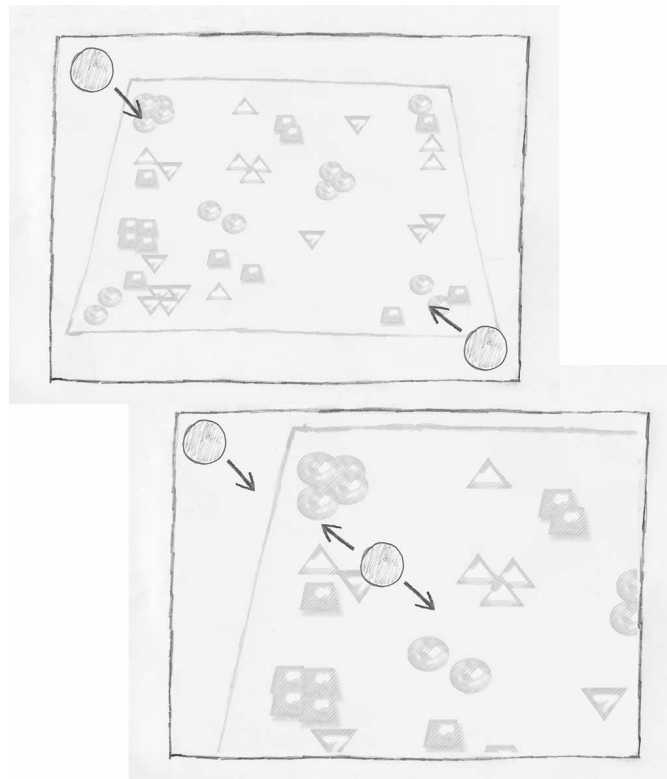


Abbildung 3.12: *Tangible Interaction* – Grafischer Entwurf für die Aktion des *Zooming*.

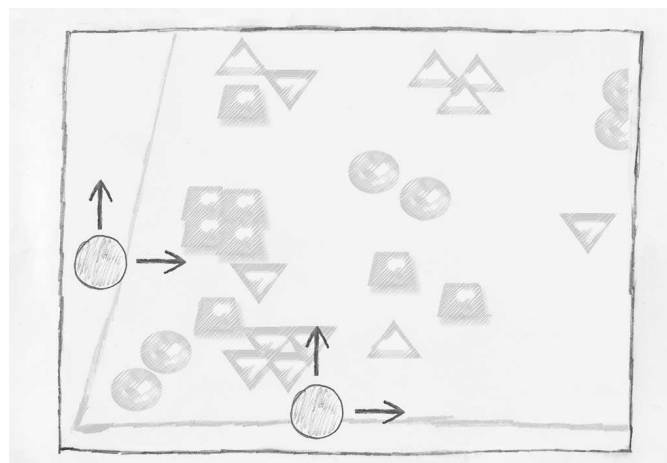


Abbildung 3.13: *Tangible Interaction* – Grafischer Entwurf für die Navigation.

Es stellte sich die Frage, wie ein Element der nun Arbeitsfläche ausgewählt werden kann. Ein erster Ansatz beschäftigte sich damit, einen aktiven Bereich zu bestimmen, der sich im Zentrum der Szenenansicht befindet und in den ein beliebiges Dateiobjekt navigiert werden muss. Nach dem Verstreichen einer gewissen Zeitspanne würde automatisch die Selektierung erfolgen. Diese Idee wurde allerdings verworfen, da die Entscheidung getroffen wurde, dass eine Aktion des Benutzers eine unmittelbare Reaktion zur Folge haben sollte, um etwaige Missinterpretationen zu verhindern. Auf Basis des Prinzips der zeitbasierten Selektierung wäre dies nicht sinnvoll gewesen. Aus diesem Grunde wurde zum Zweck der Auswahl ein zusätzliches drittes *Tangible Object* eingefügt. Dieses muss auf einem beliebigen Element des Arbeitsbereiches positioniert werden. Wird es wiederum verschoben, ist die Selektierung aufgehoben.

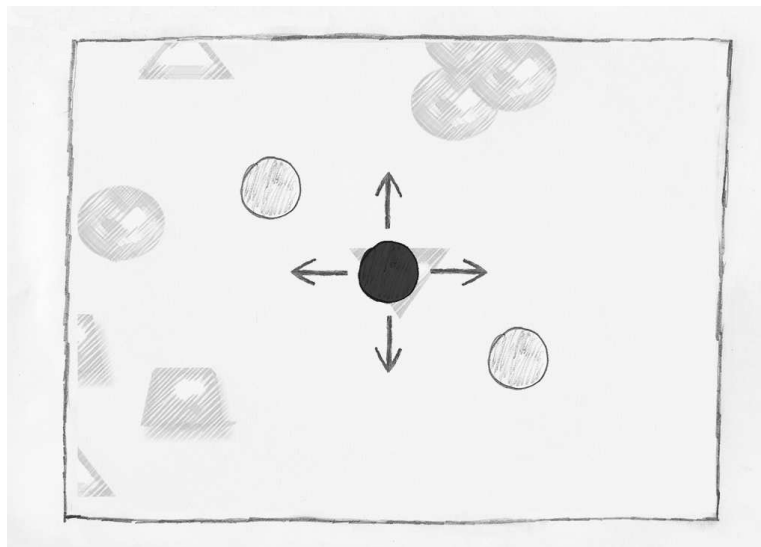


Abbildung 3.14: *Tangible Interaction* – Grafischer Entwurf für die Selektierung.

Zusammenfassend bleibt zu sagen, dass es für den Designentwurf für *Tangible Interaction* im *Zoom Space* wesentliches Kriterium war, möglichst wenige Objekte für die Interaktion zu verwenden. Dadurch sollte der Grad der Komplexität verringert und etwaigen Fehlinterpretationen bzw. nicht intendierten Verwendungen vorgebeugt werden.

Kapitel 4

Implementierung

Auf Basis der im Designprozess festgelegten Interaktionstechniken sollte im Rahmen der Implementierung eine Prototypenapplikation entstehen, in der das von Raskin beschriebene Interfacekonzept umgesetzt ist. Die primäre Herausforderung lag darin, die Designentwürfe auf Basis einer entsprechenden Programmierung und grafischen Gestaltung in die Tat umzusetzen. Die konkrete Vorgehensweise, um dieses Ziel zu erreichen, wird in diesem Kapitel beschrieben. Zusätzlich liegt ein Schwerpunkt auf der Darstellung von Problemfeldern und deren Lösungen.

Für die praktische Umsetzung wurde eine Implementierung in Kombination mit *Gesture Based Interaction* bzw. *Tangible Interaction* beschlossen. Dadurch soll die Möglichkeit geboten werden, die Vor- und Nachteile der jeweiligen Interaktionsform im Kontext des ZIP darzulegen. Zu diesem Zweck wird im Rahmen dieses Kapitels auch ein Vergleich zwischen diesen beiden Techniken durchgeführt. Als Abschluss wird ein Ausblick in die Zukunft vorgenommen, um zu zeigen, wie sich das Interaktionsparadigma der *Zooming Interfaces* auf dem Gebiet des Interaktionsdesigns etablieren und in welchen Bereichen es Anwendung finden könnte.

4.1 Systemeinsatz und Entwicklungsumgebung

Zu Beginn der Implementierungsphase stellte sich die Frage der passenden Entwicklungsumgebung. Da durch vorangegangene Projekte schon Erfahrung vorhanden war, fiel die Wahl auf *Macromedia Director*. Es handelt sich dabei um ein Autorensystem, das mehrheitlich bei der Erstellung von interaktiven Inhalten für Kiosksysteme, CD-ROMs und DVDs eingesetzt wird. Als bedeutende Funktion wird in [63] dabei die mühelose Integrationen einer Vielzahl verschiedener Medientypen angeführt. Zusätzlich besteht die Möglichkeit, 3D-Objekte zu erzeugen und in Echtzeit zu steuern. Ein beliebter Anwendungsbereich für Director 3D-Applikationen ist das Internet. Durch das so genannte *Shockwave Plug-In* bietet sich die Möglichkeit,

Director-Anwendungen in Webbrowsern abzuspielen. Man spricht in diesem Zusammenhang auch von so genannten *Shockwave-3D-Applikationen*. Zur Programmierung wird die objektorientierte Sprache *Lingo* herangezogen, die im Folgenden genauer beschrieben wird.

4.1.1 Programmiersprache Lingo

Lingo ist eine in Director integrierte Programmiersprache und wird in [21] als Event-orientiert mit objektorientierten Ansätzen beschrieben. Entsprechende Events können sowohl seitens des Programms, als auch durch den Benutzer ausgelöst werden. Darüber hinaus ist es auch möglich, Ereignisse durch den Aufruf eines so genannten *Handlers* selbst zu definieren. Diese sind vergleichbar mit jenen als Funktionen, Methoden oder Prozeduren bezeichneten Konstrukten anderer Programmiersprachen.

In den Anfängen war die Syntax von Lingo sehr stark an die natürliche englische Sprache angelehnt. Dies führte aber mitunter zu sehr langen Programmanweisungen, was sich speziell in umfangreichen Implementierungen negativ auf die Lesbarkeit des Programmcodes auswirkte. Ab Director-Version 7.0 wurde eine Erweiterung auf die so genannte DOT-Syntax durchgeführt – ein erster Schritt zur Etablierung eines objektorientierten Konzeptes. In früheren Versionen wurde der Programmcode von Director interpretiert, was aufgrund von Performancegründen geändert wurde. Lingo wird nun vor dem Programmstart kompiliert.

4.1.2 Zusatzmodul - TrackThemColors

Der Funktionsumfang von Director kann mit Hilfe von zusätzlichen Plug-Ins bedeutend erweitert werden. Diese Zusatzmodule werden als *Xtras* bezeichnet, grundsätzlich in C++ entwickelt und bieten teilweise die Möglichkeit der Integration zusätzlicher Mediendaten oder erweitern die Entwicklungsumgebung auf Lingo-Basis um weitere Module.

Ein solches Xtra mit der Bezeichnung *TrackThemColors* wurde auch für das Tracking im Rahmen der praktischen Umsetzung verwendet. Dieses bietet Funktionen bzw. Methoden, die beliebig innerhalb des Programmcodes eingesetzt werden können. Die Anwendung funktioniert folgendermaßen: ein beliebiger RGB-Wert wird an die Funktion übergeben, diese analysiert das von der Kamera gelieferte Bild und gibt die Position des entsprechenden Pixels im Bild zurück. Dadurch wird die Möglichkeit geboten, bestimmte Farbmuster im Bildverlauf zu verfolgen und damit verbunden eine Bewegung zu eruieren. Es bleibt festzuhalten, dass das Plug-In über interessante Fähigkeiten verfügt, die allerdings auch eine Vielzahl von Zusatzimplementierungen erfordern.

4.2 Grundkonzept der Bewegungserkennung

Der Schwerpunkt liegt bei der praktischen Umsetzung auf einer adäquaten Implementierung des Interfacekonzeptes der *Zooming Interfaces*. In diesem Zusammenhang spielt das interaktive System und das damit verbundene Prinzip der Bewegungserkennung (siehe Abschnitt 3.2.1) eine wichtige aber untergeordnete Rolle. Da es sich beim kamerabasierten Tracking um ein etabliertes und oftmals implementiertes Konzept handelt, wird dieses hier nur kurz am Rande behandelt.

Grundsätzlich wurde das Prinzip des Object Trackings, in Kombination mit speziellen Farbmarkern, ausgewählt. Das Kamerabild wird auf Farben hin untersucht, um in weiterer Folge Positionsdaten abzuleiten, die in der Applikation weiterverarbeitet werden können. Sämtlicher Code der in Director programmiert wurde, ist in Form von einzelnen *Scripts* organisiert, die während der Laufzeit von der Applikation entsprechend aufgerufen werden. Für das Tracking zeigen sich zwei Scripts verantwortlich, die im Folgenden analysiert werden. Zur genaueren Erklärung werden auszugsweise Exzerpte aus dem Code präsentiert.

In einem ersten Schritt werden jene Farbwerte definiert, die es zu tracken gilt. Sie werden neben anderen globalen Variablen in einem zentralen Script verwaltet. Der Benutzer wird hierzu angehalten, nach dem Start der Anwendung die Farbmarker in den entsprechenden Bereichen zu platzieren, die im Kamerabild angezeigt werden. Diese Prozedur läuft sowohl für die Variante der *Gesture Based Interaction* als auch der *Tangible Interaction* völlig identisch ab.

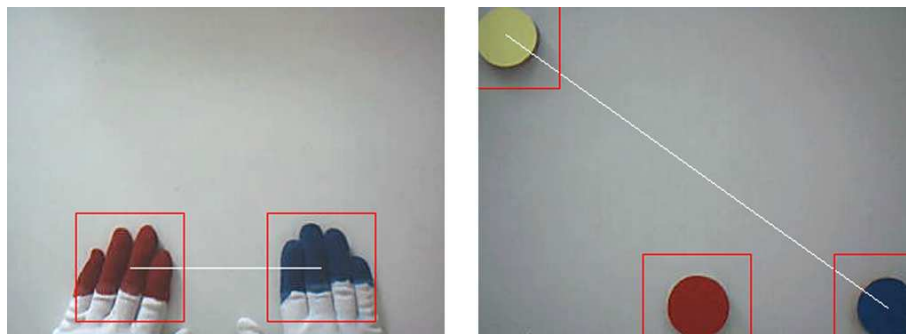


Abbildung 4.1: *Object Tracking* – Initialisierung der Farbwerte: Für die Variante der *Gesture Based Interaction* müssen zwei, für die *Tangible Interaction* drei Referenzwerte ermittelt werden.

Nach Ablauf einer Zeitspanne von fünf Sekunden werden die Koordinaten eines jeden Referenzpunktes (= Mittelpunkt des zugehörigen Rechtecks, in dem der Marker positioniert wurde) an das Plug-In *TrackThemColors* übergeben.

Grundsätzlich bieten sich nun unterschiedliche Varianten an, wie die Daten weiter verarbeitet werden können. Innerhalb der Applikation wird die Funktion der so genannten *Color-Blobs* verwendet. Dies bedeutet, dass im Bild nach jenen Bereichen gesucht wird, in denen die entsprechenden RGB-Werte am häufigsten auftreten. Innerhalb der Programmierung werden die resultierenden Daten in den beiden Arrays *yellowblob* und *blueblob* gespeichert.

```
GrabToCast(TrackObj,5)
yellowblob=TrackColorBlobs(TrackObj, trackColor1[1],
trackColor1[2],trackColor1[3],similarity, 10)

blueblob=TrackColorBlobs(TrackObj, trackColor2[1],
trackColor2[2],trackColor2[3],similarity, 10)
```

Zusätzlich werden dabei folgende Parameter an die Funktion übergeben:

- *Trackobj*: Referenz zum aktuellen Frame des Videobildes, der analysiert wird
- *trackColor [1-3]*: Selektierter Farbwert, aufgeteilt in die Komponenten [1=R], [2=G], [3=B]
- *similarity*: bestimmt den Faktor der Abweichung zwischen dem aktuellen Farbwert und jenem, der bei Initialisierung festgelegt wurde. Ist sehr hilfreich, speziell wenn sich die Lichtverhältnisse und die farbliche Beschaffenheit des Bildes ändern
- *threshold*: definiert den maximalen Abstand, der zwischen zwei Punkten auftreten darf, damit sie noch als gemeinsamer Datablob gewertet werden

Bei der Verwendung von Markern existieren mehrere Bildpunkte mit dem selben Farbwert. Dies erschwert eine genaue Positionsermittlung. Deshalb wird um die resultierende Punktwolke (*Datablob*) ein Rechteck gelegt, dessen Mittelpunkt als Referenzposition des Farbmarkers herangezogen wird. Die nachfolgend beschriebenen Prozeduren laufen für beide Farbwerte identisch ab. Aus diesem Grund wird in weiterer Folge nur der Programmcode für den gelben Farbwert präsentiert.


```

if yellowblob<>[] then
  sprite(63).rect=yellowblob[1]
  sprite(63).loch=sprite(63).loch+sprite(50).left
  sprite(63).locv=sprite(63).locv+sprite(50).top
  newyellowx=yellowblob[1].width/2+yellowblob[1].left
  newyellowy=yellowblob[1].height/2+yellowblob[1].top
end if

```

Da die Qualität des Kamerabildes großen Schwankungen unterworfen ist – was sich vor allem auf eine mangelhafte Ausleuchtung zurückführen lässt – kann es passieren, dass die farbliche Beschaffenheit nicht immer konstant ist, was entsprechende Schwankungen bezüglich der Position des Datablobs verursacht. Diese sind normalerweise zwar minimal, wirken sich aber negativ auf die Interaktion aus. Um dies zu vermeiden ist eine zusätzliche Funktion implementiert, die als *Threshold* bezeichnet wird. Damit kann sichergestellt werden, dass eine neue Referenzposition nur dann übernommen wird, wenn eine signifikante Abweichung von der vorhergehenden zu erkennen ist, die auf eine intendierte Benutzerhandlung zurückzuführen ist.

```

yellowdist=sqrt(power(oldyellowx-newyellowx,2)+
power(oldyellowy-newyellowy,2))

if(yellowdist>threshold and yellowdist<100) then
  yellowx=newyellowx
  yellowy=newyellowy
  oldyellowx=newyellowx
  oldyellowy=newyellowy
else
  yellowx=oldyellowx
  yellowy=oldyellowy
end if

```

4.2.1 Problemstellungen bei der Aufgabendurchführung

Das hier beschriebene Verfahren des Object Trackings mittels Farbmarkern geht einher mit Problemstellungen, die zum Großteil nicht auf algorithmischer Ebene zu lösen sind. Um ein zufriedenstellendes Tracking zu ermöglichen, ist die Verwendung einer qualitativ hochwertigen Kamera empfehlenswert. Zudem sollten sich die Farbmarker markant vom Hintergrund unterscheiden. Wie in den Testphasen festgestellt wurde, spielt die Ausleuchtung des Raumes, in dem das interaktive Setup aufgebaut ist, eine entscheidende Rolle. Diese sollte entsprechend hell sein und vor allem gleich bleiben.

In Bezug auf die Programmierung ist ausschlaggebend, dass jene Scripts, die als Output die aktuellen Positionen der Farbmarker wiedergeben, permanent aufgerufen werden müssen. Die von der Kamera gelieferte Bildsequenz wird an Hand der Einzelbilder analysiert, wodurch der Berechnungsaufwand mit der Höhe der Bildrate (Englisch: *Framerate*) enorm steigt. Bei einer Einstellung von fünfzehn Bildern pro Sekunde müssen sämtliche Algorithmen, die sich für das Tracken der Farben verantwortlich zeigen, für jedes Einzelbild – also fünfzehn Mal pro Sekunde – durchlaufen werden. In den Testphasen hat sich gezeigt, dass durch Einstellung einer höheren Framerate die aufgenommenen Bewegungen kontinuierlicher und flüssiger erscheinen. Zusätzlich steigt aber der Rechenaufwand erheblich, wodurch die Gesamtbelastbarkeit des Computersystems sehr schnell erreicht sein kann. Aus diesem Grund wurde die Framerate mit fünfzehn Bildern pro Sekunde beibehalten.

4.3 Aufbau der Applikation

Das Konzept einer grafischen Oberfläche, auf der verschiedene Dateiobjekte beliebig positioniert und angeordnet sind, stellt die grundlegende Idee für die Implementierung des Interfacekonzeptes der *Zooming Interfaces* dar. Sämtliche Elemente, die den Benutzer in seinem Handlungsablauf einschränken, sollen dabei auf ein Minimum reduziert werden.

Im Rahmen des Designprozesses wurden für die Gestaltung des Arbeitsbereiches und der Interaktion verschiedene Ansätze entwickelt, die eine Basis für die funktionale Implementierung darstellten. Eine wichtige Aufgabe war es dabei, eine ansprechende grafische Realisierung zu schaffen.

Grundsätzlich sind von der Applikation zwei Grundaufgaben zu erfüllen. Zum einen eine ordnungsgemäße Verarbeitung der eingehenden Tracking-Daten, zum anderen eine entsprechende Umsetzung der Benutzereingaben auf den Arbeitsbereich. Raskin beschreibt in seinen Ausführungen die Interaktion im *Zoom Space* als Tätigkeit, in der sich der Benutzer vollkommen mit seiner Aufgabe auseinandersetzt und sich auf diese fokussiert. Sämtliche Element, die eine Ablenkung verursachen können, sollen ausgeblendet werden. Aus diesem Grund ist die Applikation auf zwei Bildschirme aufgeteilt. Am primären Gerät, das sich im Fokus des Anwenders befindet, soll die grafische Darstellung des Arbeitsbereiches erfolgen. Anzumerken ist, dass die Visualisierung im konkreten Fall des in Abschnitt 3.2.1 beschriebenen interaktiven Systems auf Basis einer Projektion erfolgt.

Am sekundären Bildschirm – dieser ist in Form eines herkömmlichen Computerdisplays ausgeführt – werden ausschließlich Daten bzw. Inhalte dargestellt, die zwar zur Kontrolle des Trackings interessant sind, in weiterer Folge aber keine augenblickliche Relevanz für die Handlungen des Benutzers haben. Als Beispiel hierfür ist die Anzeige des Kamerabildes und der getrackten Farbwerte zu nennen. Zusätzlich können Konfigurationseinstel-

lungen getroffen werden, um das kamerabasierte Tracking an die äußeren Umstände anzupassen und somit dessen Genauigkeit zu erhöhen. Speziell bei einer Veränderung der Lichtverhältnisse ist eine Angleichung erforderlich. Dies kann durch eine Modifikation der Werte für Threshold und Similarity (Funktion siehe Abschnitt 4.2) erfolgen.

Durch die Schaltfläche *Key Control* wird die Möglichkeit der Interaktion mittels eines herkömmlichen Keyboards geboten. Dieser Modus ist allerdings nur für Testphasen, die im Rahmen der Implementierung permanent stattfanden, vorgesehen.

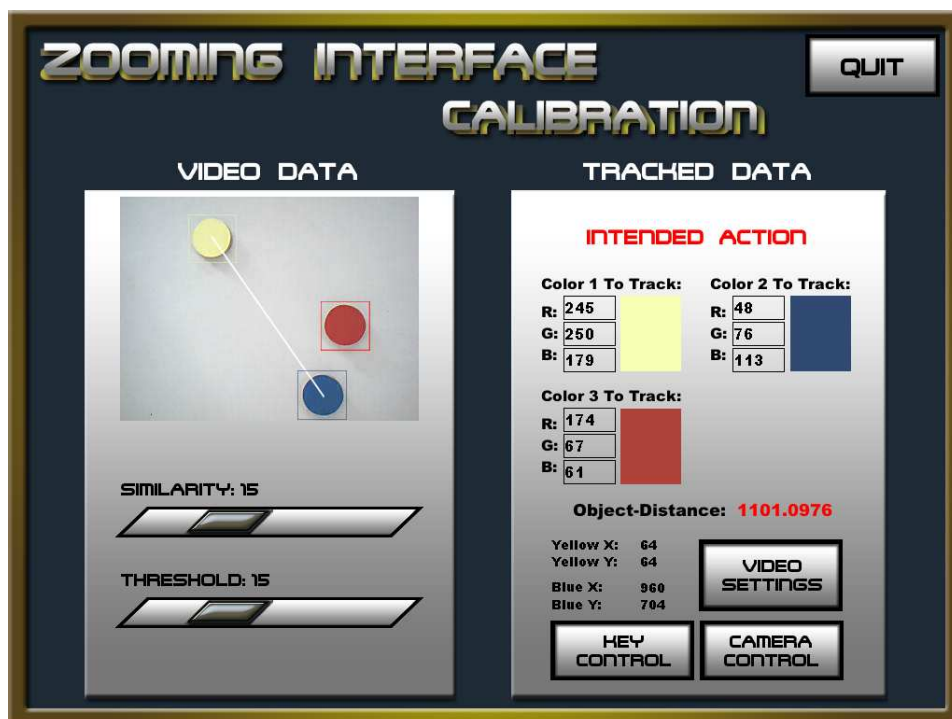


Abbildung 4.2: Applikation – Aufbau des sekundären Monitors

Für die Gestaltung der primären Anzeige wurde die in Abbildung 3.6 dargestellte Skizze als Vorbild herangezogen. Wichtig war der Aspekt der räumlichen Tiefe, der durch eine entsprechende perspektivische Verzerrung erreicht wird. Dies soll dem Benutzer zusätzlich das Prinzip der Desktop-Metapher vermitteln, indem die Interaktion auf einer ebenen Fläche passiert.

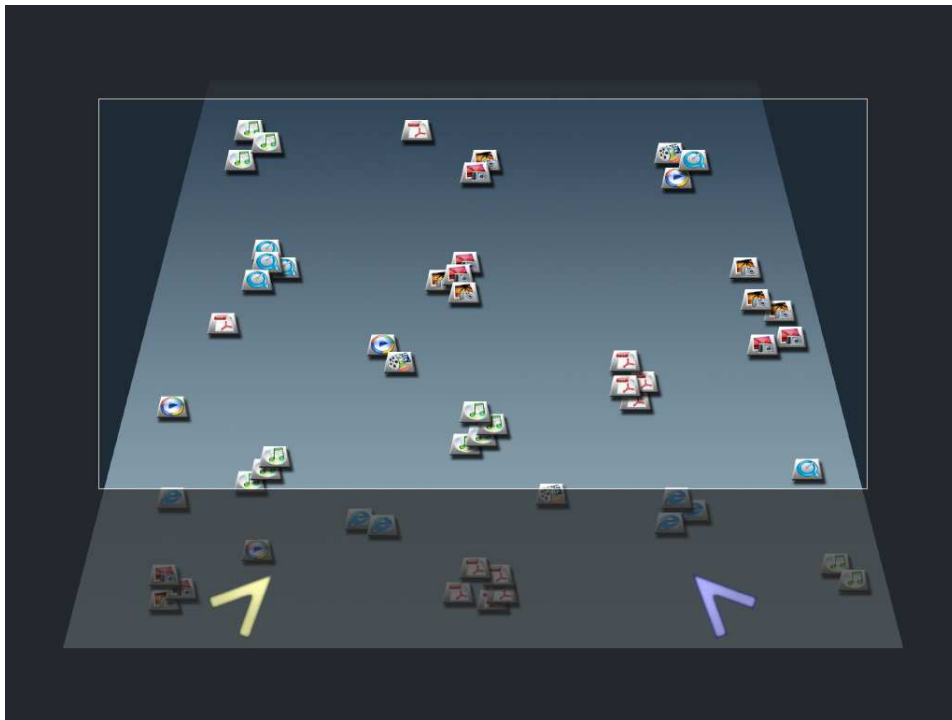


Abbildung 4.3: *Applikation* – Grafische Gestaltung der primären Anzeige, die sich im Fokus des Benutzers befindet.

Wie in Abbildung 4.3 ersichtlich, sind auf der grafischen Oberfläche verschiedene Dateitypen beliebig positioniert. Diese sind durch Piktogramme symbolisch dargestellt. Da es sich bei der entstandenen Applikation um einen Prototypen handelt, wurde nur eine geringe Anzahl von Dateiobjekten grafisch repräsentiert.



Abbildung 4.4: *Piktogramme* – Verschiedene Dateitypen werden durch Symbole dargestellt. Von links: Audio, Video, Bilder, Web, Dokumente.

Entscheidend ist, dass ein sichtbarer Zusammenhang zwischen der Abbildung und dem repräsentierten Dateiobjekt, das sich dahinter verbirgt, zu erkennen ist. Hierbei sollte man aber nicht vergessen, dass Piktogramme

selbsterklärend sein sollen, ein automatisches Verständnis aber mit einer intensiven Systemnutzung einhergeht.¹ Aus diesem Grund wurde vorrangig auf abstrakte Darstellungen gesetzt, die auch in den Betriebssystemen von Microsoft Windows XP® bzw. MAC OS X® Anwendung finden und sich daher eines gewissen Bekanntheitsgrades erfreuen. Im Rahmen der Implementierung wurde versucht, bestimmte Interaktionstechniken bzw. Prinzipien des grafischen Designs aus bereits bestehenden, etablierten Konzepten zu übernehmen. Dies sollte keine Kopie darstellen, sondern an das Konzept der Intuition anknüpfen, welches Raskin in Bezug auf Interfacegestaltung beschreibt. Die Benutzer verbinden mit natürlicher bzw. intuitiver Interaktion oftmals eine Arbeitspraxis, die sie aus ähnlichen Softwareapplikationen bzw. verwandten Tätigkeiten kennen (siehe Abschnitt 2.4.2). Würde es sich bei der hier beschriebenen Applikation um ein kommerzielles Softwareprodukt handeln, müssten die Verwendungsrechte natürlich entsprechend abgeklärt bzw. erworben werden.

4.3.1 Gesture Based Interaction im Zoom Space

In Bezug auf Interaktion mit Handbewegungen bzw. Gestiken war vor allem ein Aspekt Richtung weisend: sämtliche Aktionen, die der Benutzer setzen kann, müssen durch entsprechende Positionsverschiebungen der Hände bestimmt werden können. Das bedeutet konkret, dass nur zwei Punkte – die Position der linken Hand bzw. die Position der rechten Hand – durch das Object Tracking erkannt werden müssen. Gleichzeitig stehen aber auch nur diese zwei Punkte zur Verfügung, aus denen alle intendierten Handlungen des Benutzers abgeleitet werden müssen.

Entscheidend ist, dass die auf Basis des Trackings errechneten Werte nicht direkt für die grafische Oberfläche übernommen werden. Vielmehr werden beim Überschreiten von definierten Grenzen – durch entsprechende Positionierung der Hände im Kamerabild – nur die Werte der zugehörigen Steuervariablen geändert. Um dem Benutzer Feedback über die aktuelle Position seiner Hände im Arbeitsbereich zu vermitteln, werden diese durch entsprechende grafische Symbole (in weiterer Folge als *Handsymbole* bezeichnet) dargestellt. Diese stehen in direktem Zusammenhang mit den eingehenden Daten des Trackingsystems.

Zeigt das Kamerabild z.B. eine Handbewegung des Benutzers zum linken oberen Eckpunkt hin, so bewegt sich auch die grafische Repräsentation auf der Arbeitsfläche in diesen Bereich. Die eingehenden Daten werden also in diesem speziellen Fall 1:1 übernommen. Dem Benutzer wird Feedback vermittelt, indem die von ihm intendierte Handlung durch entsprechende Richtungsanzeigen demonstriert wird.

¹Vgl. [24]: Herczeg, 2006: S. 108

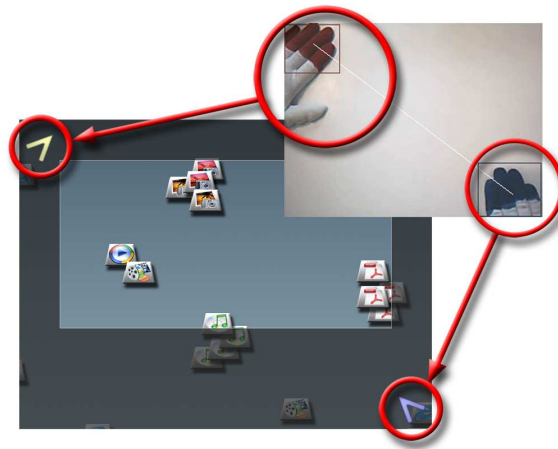


Abbildung 4.5: *Applikation* – Die eingehenden Trackingdaten werden direkt auf die Handsymbole übertragen.

In den vorangegangenen Abschnitten wurde das Prinzip des *Zoomings* als die zentrale Tätigkeit im *ZIP* beschrieben. Als maßgebliche Größe wird hierbei der Abstand zwischen den Handsymbolen herangezogen.



Abbildung 4.6: *Zooming* – Praktische Umsetzung der zentralen Interaktion im *Zoom Space*.

Im Grunde genommen entspricht das *Zooming* nur einer Skalierung der grafischen Oberfläche. Bei der Tätigkeit des *Zoom In* wird um einen fixen Wert bis zum Faktor vier vergrößert, beim *Zoom Out* wird um den selben Wert verkleinert, bis wiederum die Ausgangsgröße erreicht ist.

Die jeweilige Aktion wird durchgeführt solange eine entsprechende Distanzierung der Handsymbole vorherrscht. Dadurch wird dem Benutzer ermöglicht, den Grad der Vergrößerung bzw. Verkleinerung genau zu bestimmen. Ähnlich verhält es sich auch bei der Navigation im *Zoom Space*. Die aktiven Bereiche, in denen die Handsymbole platziert werden müssen, werden durch den äußeren, dunkleren Rahmen repräsentiert. Gemäß der Positionierung findet eine Translation der grafischen Oberfläche in die entsprechende Richtung statt. Dieser Vorgang erfolgt wiederum kontinuierlich um einen fixen Wert und zwar solange eine Überlappung zwischen den Handsymbolen und dem entsprechenden aktiven Bereich besteht.

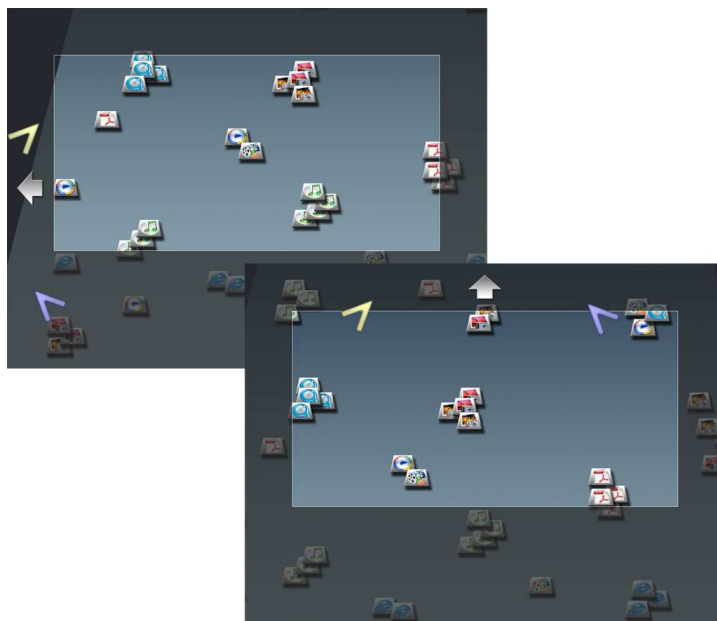


Abbildung 4.7: *Navigation* – Translation der grafischen Oberfläche durch entsprechende Positionierung der Handsymbole.

Die Rotation des Arbeitsbereiches spielt zwar in Bezug auf Interaktion im *Zooming Interface* nur eine untergeordnete Rolle, wurde aber dennoch implementiert – speziell in Hinblick auf den Einsatz des *ZIP* in kollaborativen Arbeitssituationen. Für die Berechnung des Rotationswinkels wird auf die herkömmliche Geradengleichung zurückgegriffen. Auf deren Basis kann jener Winkel berechnet werden, der der Steigung jener Geraden zwischen den Handsymbolen entspricht (vgl. Abbildung 3.10). Überschreitet bzw. unterschreitet dieser Winkel einen gewissen Wert, wird eine Drehung des Arbeitsbereiches in die entsprechende Richtung durchgeführt.

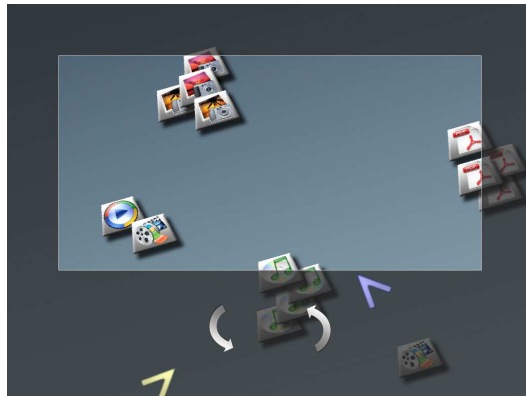


Abbildung 4.8: *Rotation* – Drehung des Arbeitsbereiches an Hand des Winkels, der sich zwischen den Handsymbolen ergibt.

Für die Selektierung eines Dateiobjektes muss es möglich sein, eine Ausführungs- bzw. Bestätigungsanweisung abzugeben, um damit den Interaktionsablauf zu vervollständigen. Wird eines der Handsymbole über einem beliebigen Dateiobjekt positioniert, kann durch ein Schließen der Hände der entscheidende Impuls ausgelöst werden. In der Programmierung wird dies an Hand einer vordefinierten Mindestdistanz überprüft, die unterschritten werden muss. Eine Objektauswahl kann nur in dem farblich helleren Bereich in der Mitte der primären Anzeige erfolgen. Hier können allerdings die Techniken *Zooming*, *Navigation* und *Rotation* nicht angewendet werden, um den Ablauf der Selektierung nicht zu komplizieren.

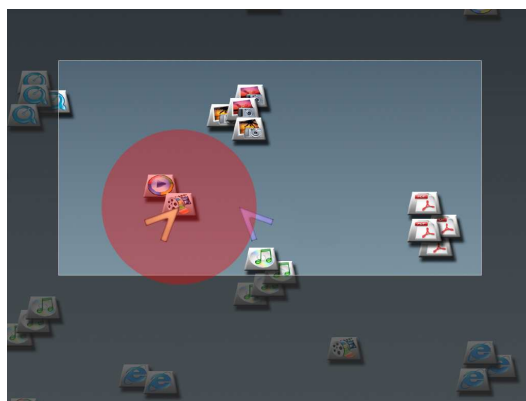


Abbildung 4.9: *Selektierung* – Befinden sich beide Handsymbole in einem gemeinsamen Auswahlkreis, so wird ein Impuls ausgelöst.

Als Resultat der Auswahl wird das Interface dem Dateitypen entsprechend angepasst und bietet adäquate Möglichkeiten der Bearbeitung. So beschreibt es zumindest Raskin in seinen Ausführungen. Da innerhalb der praktischen Umsetzung der Schwerpunkt auf der Implementierung der passenden Interaktionstechniken für das *ZIP* lag, wurde auf die Ausführung konkreter Anwendungen, gemäß den ausgewählten Dateitypen, verzichtet. Dies stellt aber sehr wohl eine interessante Ausgangsposition für zukünftige Erweiterungen dar. Grundsätzlich ist dieser Fall in der aktuellen Implementierung folgendermaßen gelöst: Nach der Selektierung eines Dateiobjekts erscheint eine grafische Oberfläche, auf der Videos, Bilder, Audiodateien, Dokumente oder Websites präsentiert bzw. abgespielt werden. Die Abwahl bzw. Schließung der Oberfläche erfolgt durch Selektieren der Close-Schaltfläche, symbolisiert durch ein *X*.

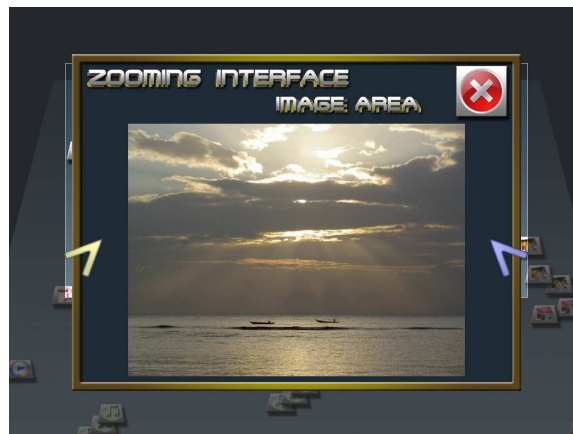


Abbildung 4.10: *Applikationsbereich* – Hier am Beispiel der Anzeige eines Fotos in der *Image Area*.

Zusammenfassend bleibt zum Thema *Gesture Based Interaction* im *Zoom Space* zu sagen, dass sämtliche Transformationen der grafischen Oberfläche, also Skalierung, Translation und Rotation, auf Basis der Position zweier Punkte durchgeführt werden können. Dies stellte im Rahmen der Implementierung eine gewisse Herausforderung dar. In Bezug auf das Prinzip der Objektselektierung durch das Schließen der Hände, ergab sich eine Problemstellung, deren Lösung in Abschnitt 4.3.3 erörtert wird.

4.3.2 Tangible Interaction im Zoom Space

Ein richtungweisender Aspekt in Bezug auf *Tangible Interaction* ist die direkte Verknüpfung realer Objekte mit virtuellen Daten (vgl. Abschnitt 2.6.3). Im Rahmen des Designkonzeptes wurde festgelegt, dass bei dieser Form der Interaktion im *Zoom Space* sämtliche Aktionen des Benutzers

durch Manipulation von drei Objekten durchgeführt werden können. Die auf Basis des Trackings errechneten Werte werden dabei direkt für die grafische Oberfläche übernommen, was eine entsprechende Genauigkeit der Daten erfordert. Jene Objekte die zur Interaktion herangezogen werden, sind innerhalb der Anwendung durch Abbildungen repräsentiert, um so dem Benutzer einen Eindruck über die aktuellen Positionen zu vermitteln.

Das Prinzip des *Zooming* ist in der Programmierung in Form einer permanenten Skalierung der grafischen Oberfläche umgesetzt. Um einen entsprechenden Faktor zu berechnen, wird jenes Rechteck herangezogen, welches zwischen den beiden Navigationsobjekten gebildet werden kann. Dieses ist durch eine rote Fläche dargestellt. Beim Starten der Applikation befinden sich die Objekte an den jeweiligen äußeren Eckpunkten, sodass die Distanz zwischen den beiden maximal ist, und auch die größtmögliche Fläche gebildet wird. Der errechnete Skalierungsfaktor ist in diesem Fall 1.0. Bewegt man die Objekte nun aufeinander zu, wodurch die Distanz zwischen diesen verringert wird, so reduziert sich zwar der Flächeninhalt, indirekt proportional dazu erhöht sich aber der Skalierungsfaktor und die grafische Oberfläche wird vergrößert. Wird im umgekehrten Fall die Distanz zwischen den Objekten erhöht, so hat dies wiederum eine Verkleinerung zur Folge.

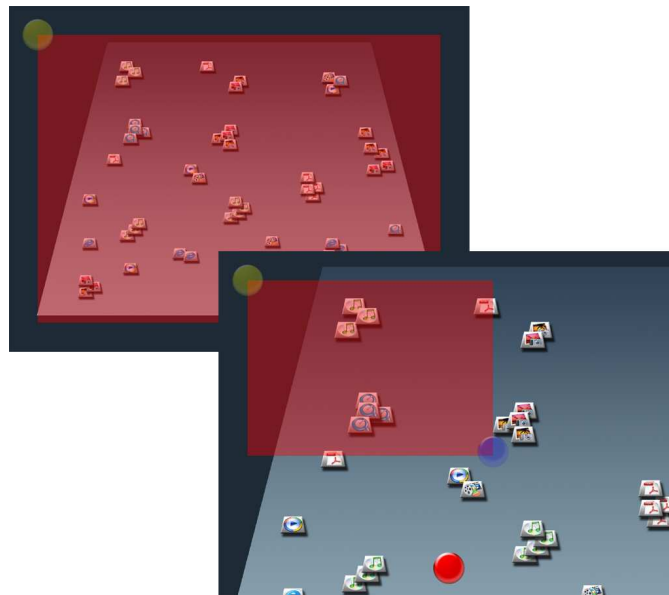


Abbildung 4.11: *Zooming* – Die grafische Oberfläche wird dem errechneten Skalierungsfaktor entsprechend vergrößert bzw. verkleinert.

Die Navigation im Arbeitsbereich ist realisiert auf Basis einer Translation der grafischen Oberfläche. In der Programmierung wird zu diesem Zweck der Mittelpunkt des Rechtecks berechnet, welches von den Navigationsob-

jekten gebildet wird. Werden sie verschoben, erfolgt eine Veränderung des Punktes in entsprechendem Ausmaß. Die Distanz zwischen der aktuellen und der vorhergehenden Position wird für eine Verschiebung der grafischen Oberfläche herangezogen.

Die Selektierung eines Dateiobjektes erfolgt – wie im Designkonzept festgelegt – durch ein drittes *Tangible Object*. Dies konnte innerhalb der Software einfach umgesetzt werden, da nur auf eine Überlappung des Auswahlobjektes und eines Dateiobjektes hin überprüft werden muss. Im Gegensatz zur *Gesture Based Interaction* wird die Abwahl nicht durch ein explizites Schließen des Applikationsbereiches erreicht, sondern durch eine erneute Verschiebung des Auswahlobjektes. Aus diesem Grund wird die zugehörige Applikation nicht im Arbeitsbereich, sondern am sekundären Computerdisplay geöffnet. Dadurch wird sichergestellt, dass – gemäß den Richtlinien der *Tangible Interaction* – der Kontext zwischen den Objekten, die zur Navigation herangezogen werden, erhalten bleibt. Dieser Sachverhalt wird in Abschnitt 4.4 noch genauer analysiert.



Abbildung 4.12: *Navigation und Selektierung* – Bei einer Überlappung zwischen Auswahlobjekt und Dateiobjekt öffnet sich der Applikationsbereich

Für die *Tangible Interaction* im *Zoom Space* werden drei Objekte verwendet, von denen zwei ausschließlich zur Skalierung bzw. Navigation genutzt werden und eines zur Selektierung herangezogen wird. Dies soll vor allem dazu dienen, die Interaktion im *Zooming Interface* für den Benutzer so einfach wie möglich zu gestalten, da die *Tangible Objects* leicht differenziert und ihrer Funktion entsprechend zugeordnet werden können.

4.3.3 Problemstellungen bei der Aufgabendurchführung

Bei der Implementierung des *ZIP* ergaben sich in erster Linie bei der Umsetzung und Integration der beiden Interaktionstechniken verschiedene Problemstellungen. Diese waren teilweise durch eine Modifikation der Programmierung oder durch eine Änderung auf Hardware-Ebene zu lösen.

Im Falle der *Gesture Based Interaction* wurde im Rahmen der Testphasen die Erkenntnis gewonnen, dass bei einer Verwendung von Handschuhen als Farbmarker die Genauigkeit des Trackings nicht ausreichend war. In Abschnitt 4.2 wurde das Konzept der Datablobs beschrieben: wenn mehrere Bildpunkte mit den selben Farbwerten auftreten, wird um die gesamte Punktewolke ein Rechteck gelegt, dessen Mittelpunkt als Referenzpunkt für die Interaktion des Benutzers herangezogen wird. Da die verwendeten Handschuhe eine große Fläche besitzen, befand sich der, aus dem resultierenden Datablob errechnete, Referenzpunkt ungefähr in der Mitte der Handfläche, was für eine Interaktion mittels Gestiken nicht genau genug war, da durch das Zeigen auf ein Objekt eine Verbindung zu diesem hergestellt wird (siehe Abschnitt 2.6.5). Es war daher nahe liegend die verwendeten Farbmarker zu verkleinern. Diese sollten aber weiterhin an den Händen getragen werden können. Deshalb wurden in weiterer Folge weiße Handschuhe eingesetzt, die nur mehr im Bereich der Fingerspitzen eingefärbt sind. Die weitaus kleinere Fläche war für ein ordnungsgemäßes Tracking völlig ausreichend. Durch diese Maßnahme konnte die Genauigkeit der Benutzerinteraktion bedeutend erhöht werden.



Abbildung 4.13: *Gesture Based Interaction* – Modifizierte Farbmarker

Eine weitere Problemstellung, die wiederum bei der *Gesture Based Interaction* auftrat, ergab sich bei der Auswahl eines Dateiobjektes und der nachfolgenden Schließung des zugehörigen Applikationsbereiches. Durch entspre-

chende Navigation der grafischen Oberfläche konnte der Fall auftreten, dass sich die Close-Schaltfläche des Applikationsbereiches direkt über einem Dateiobjekt befand. Dadurch wurde ein permanentes und endloses Öffnen bzw. Schließen ausgelöst, was zum Absturz der Applikation führen konnte.

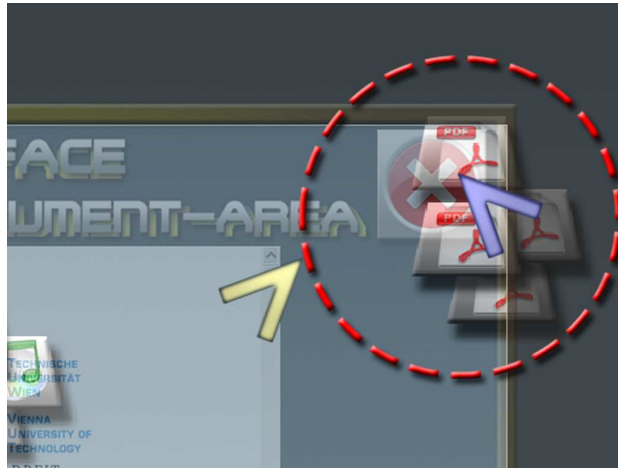


Abbildung 4.14: *Gesture Based Interaction* – Entstehung des *Öffnen-Schließen-Problems* bei der Objektauswahl

Um dies zu verhindern wurde der entscheidende Impuls, der durch das Zusammenführen der Hände ausgelöst wird, an eine definierte Zeitspanne gebunden. Erst zwei Sekunden nach Selektierung der Close-Schaltfläche ist eine erneute Objektauswahl möglich. Dieser Zeitraum wurde dabei als ausreichend angesehen, um eine entsprechende Navigation im Arbeitsbereich durchzuführen.

In Bezug auf *Tangible Interaction* stellte sich vor allem der Umstand der direkten Datenübertragung vom Trackingsystem auf die grafische Oberfläche als problematisch heraus. Grundsätzlich treten bei einer optimalen Funktion des Systems keine nennenswerten Probleme auf. Da das Prinzip des Object Trackings allerdings von äußeren Faktoren abhängig ist, kann sich z.B. eine Veränderung der Ausleuchtung negativ auf das Tracking auswirken, was eine ordnungsgemäße Interaktion beinahe unmöglich macht. Ein Lösungsansatz für dieses Problem ist z.B. in einer verbesserten Feinabstimmung des Trackingsystems, durch passende Einstellung der Parameter Similarity und Threshold (siehe Abschnitt 4.2), zu finden. Zusätzlich wäre eine Modifikation auf Ebene der Programmierung denkbar, indem die eingehenden Trackingdaten nicht direkt auf die grafische Oberfläche übertragen, sondern in einem vorhergehenden Prozess gefiltert werden. Somit würden Skalierung und Translation gleichmäßiger erfolgen, wodurch die Sensibilität des Systems verringert und die Interaktion für den Benutzer verbessert werden könnte.

In Zusammenhang mit den hier beschriebenen Problemstellungen, lässt sich wiederum auf die Notwendigkeit von intensiven Testphasen hinweisen, die im Rahmen der praktischen Umsetzung durchgeführt wurden. Viele Problemherde könnten sonst nur schwer erkannt werden.

4.4 Vergleich: Tangible vs. Gesture Based Interaction

Wie aus dem vorangegangenen Abschnitt hervorgeht, ergeben sich bei einer ordnungsgemäßen Implementierung des *ZIP* in Kombination mit *Tangible* bzw. *Gesture Based Interaction* diverse Problemstellungen, die es zu lösen gilt. Zusätzlich stellt sich die Frage, welche der beiden Interaktionsformen sich für eine Interaktion im *Zooming Interface* eignet. Um diese zu beantworten, müsste man an die bereits getane Arbeit eine Feldforschung anknüpfen, in der Testreihen gemeinsam mit verschiedenen Benutzergruppen durchgeführt werden. Dadurch würden sich die Stärken und Schwächen des *ZIP* genau analysieren lassen. Dies könnte eine interessante Thematik für zukünftige Arbeiten in diesem Bereich darstellen.

Innerhalb dieser Arbeit beschränkt sich die Evaluierung der beiden Interaktionsformen auf einen einfachen Designtest. Als Beteiligte fungierten drei Kollegen, die ihrerseits Studien im Bereich der Informatik absolvieren. Im Rahmen eines Testaufbaus wurden sie angehalten, sich auf Basis einer kurzen Einführung mit dem Setup auseinanderzusetzen. Dezidierte Aufgabenstellung gab es keine. Jeder Benutzer sollte nach Belieben den Arbeitsbereich erkunden, um nach Abschluss seiner Tätigkeit eine kurze Erklärung abzugeben, welche Interaktionsform als passender eingestuft wird und welche Aspekte für diese Entscheidung ausschlaggebend sind. Das Ergebnis fiel eindeutig aus. Alle drei Benutzer waren der Meinung, dass sich die *Gesture Based Interaction* besser zur Interaktion im *Zoom Space* eignet. Als interessanter Aspekt wurde die Möglichkeit der Interaktion mittels Handbewegungen genannt, was sich beim Modus der Selektierung durch das *Zeigen auf ein Objekt* manifestiert.

Als Hauptgrund für die Ablehnung der *Tangible Interaction* wurde die starke Verknüpfung der *Tangible Objects* und der grafischen Oberfläche genannt. Das *Zooming* und die Navigation im Arbeitsbereich konnten zwar ohne weiteres durchgeführt werden, allerdings liefen diese Tätigkeiten bei weitem nicht so konstant und homogen ab wie z.B. bei der Interaktion mittels Gestiken. Die Selektierung eines Dateiobjektes bereitete geringere Probleme. Die direkte Verbindung zwischen realem Auswahlobjekt und der virtuellen Repräsentation wurde nicht als hinderlich eingestuft.

Diese Erkenntnisse waren durchaus kongruent mit den persönlichen Erfahrungen, die im Rahmen der Implementierung gewonnen werden konnten. Die Vorliebe zur Interaktion mittels Gestiken lässt sich dabei auf Basis der

Programmierung erklären. Ein ausschlaggebender Punkt ist, dass die vom Trackingsystem gelieferten Daten nicht direkt auf die grafische Oberfläche, sondern auf die Handsymbole umgesetzt werden. Erst durch deren Positionierung in den aktiven Bereichen wird eine entsprechende Aktion ausgelöst. Dies kann vom Benutzer sehr gut beeinflusst werden, wodurch er permanent volle Kontrolle über den Arbeitsbereich hat.

Im Fall der *Tangible Interaction* ergeben sich die Probleme durch die direkte Verknüpfung der eingehende Trackingdaten und der Transformation der grafischen Oberfläche. Das gesamte System reagiert deshalb sehr sensibel auf die Benutzereingaben. Vor allem die Navigation im Arbeitsbereich wurde in diesem Zusammenhang von den drei Benutzern als schwierig angesehen. Es ist beinahe unmöglich, beide Navigationsobjekte in eine Richtung zu verschieben ohne die Distanz zwischen diesen zu verändern. Dadurch wird im Moment der Navigation eine Skalierung durchgeführt, die nicht beabsichtigt ist. Eine Lösung für dieses Problem ist in der Filterung der eingehenden Trackingdaten zu finden (siehe Abschnitt 4.3.3).

Zusätzlich ist klarzustellen, dass das Konzept der *Tangible Interaction* an strikte Grundrichtlinien gebunden ist. Ein entscheidender Aspekt ist der Kontext zwischen den verschiedenen realen *Tangible Objects* und deren virtuellen Pendanten, der nicht gebrochen werden soll. Dieses Problem wird deutlich bei der Auswahl eines Dateiobjektes und der weiterführenden Bearbeitung im Applikationsbereich. Der gesamte Interaktionsprozess wird dabei von der Ebene der Navigation und Selektierung auf die Ebene der Manipulation transferiert. Hier wären nun zusätzliche Objekte zur Interaktion notwendig, da jene, die bereits in Verwendung sind, in einem definierten Kontext eingesetzt werden, aus dem sie nicht herausgerissen werden dürfen. Ansonsten würde dies nicht den Richtlinien der *Tangible Interaction* entsprechen. In der in Abschnitt 4.3.2 beschriebenen Implementierung stellte sich das Problem nicht unmittelbar, da im Rahmen vorliegender Arbeit die Applikationsbereiche nur rudimentär ausgeführt sind. Diese öffnen sich dabei am sekundären Monitor, wodurch das Problem des Wechsels der Interaktionsebene temporär gelöst werden konnte.

Zusammenfassend bleibt zu sagen, dass der hier beschriebene Evaluierungsprozess dem Zweck diene eine Reaktion von Dritten einzuholen, um im Zuge des Vergleiches zwischen den beiden implementierten Interaktionstechniken eine Argumentationsbasis zu definieren. Grundsätzlich konnte die Erkenntnis gewonnen werden, dass die *Tangible Interaction* mit komplexen Problemstellungen verbunden ist, die nach einer Lösung verlangen. Es stellt sich zudem die Frage, ob die Interaktion des *Zooming* – durch eine Abgrenzung der Ansicht auf den Arbeitsbereich – mit den Ansätzen von Jef Raskin, der ja von einer grenzenlosen, universalen Oberfläche spricht, kombiniert werden kann. Aus diesem Grund erscheint die *Gesture Based Interaction* zum aktuellen Zeitpunkt als die passende Interaktionsform im *Zooming Interface*.

Kapitel 5

Schlussbemerkungen

Die Konzeption und Gestaltung interaktiver Benutzerschnittstellen ist eine bedeutende Herausforderung und stellt permanent neue Anforderungen an die beruflichen Gruppen, die sich mit dieser Thematik auseinandersetzen. Dieser Prozess erfordert detailliertes Spezialwissen und bindet Expertisen aus verschiedenen Fachrichtungen mit ein. Interaktionsdesign wird dabei zu einer wesentlichen und maßgeblichen Tätigkeit.

In der vorliegenden Arbeit wurde versucht, einen Überblick über die gängigen Prinzipien der Benutzerinteraktion zu vermitteln. Es existieren sehr viele unterschiedliche Ansätze der Kommunikation zwischen Mensch und Maschine. Diese wurden als so genannte Interaktionsparadigmen bezeichnet. Eine Präsentation dieser Modelle und Konzepte, stellt den Inhalt des ersten Teiles dieser Arbeit dar. Eine konkrete Analyse der Ergebnisse wird in Abschnitt 2.7 vorgenommen, weshalb hier nicht im Detail darauf eingegangen wird. Eine wichtige Erkenntnis war, dass der Begriff des Interaktionsparadigmas nur schwer im Detail zu beschreiben ist. Mehrheitlich wird er verwendet, um System- bzw. Kommunikationsmodelle zu beschreiben oder zu definieren.

Ein erklärtes Ziel dieser Arbeit war es, auf Basis der Ergebnisse der Literaturrecherche ein Interaktionsparadigma zu finden, das sich auch in einer interessanten Implementierung praktisch umsetzen lässt. Hierbei fiel die Wahl auf die Modelle und Ideen von Jef Raskin. Das von ihm kreierte Konzept der *Zooming Interfaces* ist ein interessanter Designansatz, der innovative Aspekte beinhaltet. Die Informationen sind in Datenblöcken organisiert, die auf einer beliebig großen Fläche verteilt sind, auf der man frei navigieren kann. Durch die elementare Aktion des *Zoomings* wird es möglich, jene Teilbereiche, die nur schwer zu erkennen sind im Detail zu betrachten, um damit verbunden die unterschiedlichen Informationsebenen zu erkunden. Raskin sieht dieses Konzept der Interaktion als Erweiterung zu den herkömmlichen Desktop-basierten Systemen.

Im Rahmen der praktischen Implementierung wurde Raskin's Idee in Form einer grafischen Oberfläche umgesetzt, auf der verschiedene Dateiobjekte positioniert sind. Nach Auswahl eines Objekts passt sich das Interface dem entsprechenden Typus an und bietet adäquate Optionen zur Bearbeitung und Manipulation. Zur Interaktion wurden die Prinzipien der *Gesture Based* bzw. *Tangible Interaction* herangezogen, was die Konzeption eines interaktiven Systems erforderte. Hierzu wurde ein kamerabasiertes Trackingsystem konstruiert, um die Bewegungen und Gestiken bzw. Positionsverschiebungen der *Tangible Objects* analysieren zu können. Dadurch wird dem Benutzer die Möglichkeit der Interaktion mit einer Softwareapplikation geboten.

In einer abschließenden Gegenüberstellung wurde festgestellt, dass sich die *Gesture based Interaction* sehr gut eignet für eine Interaktion im *Zoom Space*. Die *Tangible Interaction* ist mit einigen Problemstellungen verbunden, die schwer zu lösen sind (siehe Abschnitt 4.4).

Die praktische Umsetzung gestaltete sich zwar zeitaufwändig, wurde aber mit großem Enthusiasmus durchgeführt. Hauptgrund hierfür war das interessante Konzept der *Zooming Interfaces*. Raskin führt in [46] wenige praktische Beispiele dafür an, was aber auch positiv ist, da er keine möglichen Einsatzbereiche vorwegnimmt oder ausschließt. Daher war es möglich, der Kreativität freien Lauf zu lassen, wodurch ein Interaktionskonzept entstand, das einen innovativen Charakter besitzt.

Abschließend ist festzuhalten, dass Interaktionsdesign eine sehr umfangreiche Tätigkeit ist, die wirkliches Fachwissen in verschiedenen Gebieten erfordert. Die Gestaltung interaktiver Systeme verlangt nach Lösungen für vielseitige Problemstellungen, die aus unterschiedlichen fachlichen Perspektiven gesehen werden können. Das gewählte Thema der Interaktionsparadigmen stellt eine interessante Thematik dar. In Kombination mit den Ergebnissen der praktischen Umsetzung ist eine Masterarbeit entstanden, die aus persönlicher Sicht als Erfolg zu betrachten ist. Zum einen konnte gewisses Fachwissen in Bezug auf Interaktionsdesign gewonnen werden, zum anderen wurde das gesteckte Ziel – die Implementierung eines interaktiven Systems – erfüllt. Es bestehen aber noch Verbesserungsmöglichkeiten, die im folgenden Ausblick auf die zukünftige Arbeit angesprochen werden.

5.1 Ausblick für die Zukunft

In den vorangegangenen Abschnitten wurden schon diverse Szenarien erwähnt, die sich mit möglichen zukünftigen Entwicklungen beschäftigen, die an das Konzept der *Zooming Interfaces* anknüpfen. Bezogen auf die praktische Umsetzung, die in dieser Arbeit beschrieben wurde, wird sich eine weiterführende Implementierung vorrangig auf das Prinzip der *Gesture Based Interaction* beschränken. Aus diesem Grund werden hier nur Ansätze dis-

kutiert, die sich mit diesem Konzept auseinandersetzen.

Im Rahmen dieser Arbeit galt die Umsetzung des Navigations- und Selektierungsinterfaces als Grundanforderung. Was die Implementierung der Applikationsbereiche betrifft, wäre großes Potential für eine fortführende Entwicklung gegeben. Grundsätzlich wäre es interessant für jeden Dateityp individuelle Methoden der Bearbeitung bzw. direkten Manipulation anzubieten. Dadurch würde sich das Potential des Interfaces auf funktionaler Ebene bedeutend erhöhen.

Ein weiterer Ansatz beschäftigt sich mit der Interaktion im Arbeitsbereich. Zum aktuellen Zeitpunkt sind sämtliche Dateiobjekte an definierte Positionen gebunden. Hierbei wäre es dienlich, zusätzliche Optionen der beliebigen Platzierung und Gruppierung anzubieten. Dadurch wären die Benutzer in der Lage, den Arbeitsbereich entsprechend zu adaptieren, um eine Abstimmung, gemäß den persönlichen Vorlieben, durchzuführen.

Raskin geht in seinen Ausführungen auch auf den Einsatz in kollaborativen Arbeitssituationen als mögliches Anwendungsszenario ein. In Abschnitt 4.3.1 wurde darauf hingewiesen, dass durch Implementierung des Rotationsmodus schon ein erster Schritt in diese Richtung getan wurde. Hierbei besteht aber noch großes Entwicklungspotential, z.B. in Form von geteilten Arbeitsbereichen in Kombination mit Möglichkeiten des Datenaustausches. Abschließend bleibt zu sagen, dass für eine ordnungsgemäße Analyse bzw. Evaluierung des Konzeptes der *Zooming Interfaces* entsprechende Feldforschung notwendig ist. Dies könnte nicht nur nützlich sein, um potentielle Schwachstellen aufzudecken und zu eliminieren, sondern auch eine Möglichkeit bieten, zukünftige Einsatzgebiete zu definieren.

Anhang A

Inhalt der CD-ROM

File System: ISO9660 Level 2

Mode: Single-Session (CD-ROM)

A.1 Kurzfassung / Abstract

Pfad: /documents/

kurzfassung.pdf Kurzfassung (PDF-File)

abstract.pdf Abstract (PDF-File)

A.2 Masterarbeit

Pfad: /documents/

masterarbeit.pdf Masterarbeit (PDF-File)

A.3 Software

Pfad: /software/

zooming_interface_GB.exe (Exe-File)

zooming_interface_Tl.exe (Exe-File)

A.4 Videos

Pfad: /videos/

zooming_interface_GB.avi (AVI-File)

zooming_interface_Tl.avi (AVI-File)

Literaturverzeichnis

- [1] ANIRUDH, J. und S. RASAL: *Interaction Paradigm for Semi-Literates and Illiterates*. Technischer Bericht, Industrial Design Center Mumbai, 2002.
- [2] BASTIDE, R.: *Engineering Human-Computer Interaction and Interactive Systems – Joint Working Conferences*. Springer-Verlag, 2005.
- [3] BEIGL, M.: *Point and Click – Interaction in Smart Environments*. Technischer Bericht, University of Karlsruhe, 1999.
- [4] BELLOTTI, V., M. BACK, K. EDWARDS, R. GRINTER, A. HENDERSON und C. LOPES: *Making Sense of Sensing Systems – Five Questions for Designers and Researchers*. In: *Proceedings CHI 2002, CHI Letters, Volume 1 of 1*, S. 415–422, Minneapolis, USA, 20-25 April 2002. ACM.
- [5] BENFORD, S., H. SCHNADELBACH, B. KOLEVA, B. GAVER, A. SCHMIDT, A. BOUCHER, S. ANTHONY, R. ANASTASI, C. GREENHALGH, T. RODDEN und H. GELLERSEN: *Sensible, Sensable and Desirable – A Framework for Designing Physical Interfaces*. Technischer Bericht, Equator Interdisciplinary Research Collaboration, 2003.
- [6] BENGIO, S.: *Machine learning for Multimodal Interaction*. Springer-Verlag, 2005.
- [7] BOHN, J. und M. ROHS: *Klicken in der realen Welt – Workshop Mensch-Computer-Interaktion in allgegenwärtigen Informationssystemen*. Technischer Bericht, ETH Zürich, 2001.
- [8] BORCHERS, J.: *Interaction Design Patterns – Twelve Theses*. Technischer Bericht, University of Linz, 2000.
- [9] BREWSTER, S.: *Haptic Human-Computer Interaction*. Springer-Verlag, 2001.
- [10] BUNT, H., R.-J. BEUN und B. TIJN: *Multimodal Human-Computer Communication – Systems, Techniques and Experiments*. Springer-Verlag, 1998.

- [11] BURBRIDGE, M.: *Bildquelle*. URL, www.michaelburbridge.com/design/capabilities.php, 04. Jan.: 13:14, 2007.
- [12] CAMARATA, K., E. YI-LUEN DO, B. JOHNSON und M. GROSS: *Navigating Information Space with Tangible Media*. Technischer Bericht, University of Washington, Seattle, 2002.
- [13] CAMURRI, A.: *Gesture-based Communication in Human-Computer Interaction*. Springer-Verlag, 2004.
- [14] CARROLL, J.: *Designing Interaction – Psychology at the Human-Computer Interface*. Cambridge University Press, 1991.
- [15] CARROLL, J.: *Human-Computer Interaction in the New Millennium*. ACM Press, 2002.
- [16] CHEN, Q.: *Human Computer Interaction – Issues and Challenges*. Idea Group Publishing, 2001.
- [17] DAHM, M.: *Grundlagen der Mensch-Computer-Interaktion*. Pearson Studium, 2006.
- [18] DIX, A.: *Human-Computer Interaction*. Prentice-Hall, 2002.
- [19] EISENSTEIN, J. und W. MACKAY: *Interacting with Communication Appliances - An evaluation of two computer visionbased selection techniques*. Technischer Bericht, Massachusetts Institute of Technology and INRIA Futurs, Université Paris-Sud, 2006.
- [20] FORBRIG, P.: *Interactive Systems: Design, Specification, and Verification – 9th International Workshop*. Springer Verlag, 2002.
- [21] GROSS, P. und M. GROSS: *Macromedia Director 8.5 Shockwave Studio für 3D*. Markt+Technik Verlag, 2002.
- [22] HEINECKE, A.: *Mensch-Computer-Interaktion*. Carl Hanser Verlag, 2004.
- [23] HERCZEG, M.: *Software-Ergonomie*. Oldenbourg Wissenschaftsverlag, 2005.
- [24] HERCZEG, M.: *Interaktionsdesign – Gestaltung interaktiver und multimedialer Systeme*. Oldenbourg Wissenschaftsverlag, 2006.
- [25] INKPEN, K., K. BOOTH und M. KLAWE: *Drag-and-Drop vs. Point-and-Click – Mouse Interaction for Children*. Technischer Bericht, University of British Columbia, Vancouver, 1996.

- [26] ISHII, H., M. ORTH und M. GORBET: *Triangles – Tangible Interface for Manipulation and Exploration of Digital Information Topography*. Technischer Bericht, Massachusetts Institute of Technology, 1998.
- [27] ISHII, H. und B. ULLMER: *The metaDESK – Models and Prototypes for Tangible User Interfaces*. Technischer Bericht, Massachusetts Institute of Technology, 1997.
- [28] ISHII, H. und B. ULLMER: *Tangible Bits – Towards Seamless Interfaces*. Technischer Bericht, M.I.T. Media Laboratory, 1997.
- [29] JACKO, J.: *The Human-Computer Interaction Handbook – Fundamentals, Evolving Technologies and Emerging Applications*. New York, 2003.
- [30] JOHNSON, C.: *Interactive Systems – Design, Specification, and Verification*. Springer-Verlag, 2001.
- [31] LAUREL, B.: *The art of Human-Computer Interface Design*. Addison-Wesley, 1999.
- [32] LI, J., C. PLAISANT und B. SHNEIDERMAN: *Data Object and Label Placement For Information Abundant Visualizations*. Technischer Bericht, University of Maryland, 1998.
- [33] LOGOTRON: *Bildquelle*. URL, www.logo.com/cat/view/volito-tablet.html, 04. Jan.: 12:48, 2007.
- [34] MANDRYK, R., K. INKPEN, M. BILEZIKJIAN, S. KLEMMER und J. LANDAY: *Exploring a New Interaction Paradigm for Collaborating on Handheld Computers*. Technischer Bericht, Simon Fraser University, California, 2000.
- [35] MAQUIL, V.: *Tangible Interaction in Mixed Reality Applications*. Diplomarbeit, University of Technology, Vienna, 2006.
- [36] MCGRENERE, J., R. BAECKER und K. BOOTH: *An Evaluation of a Multiple Interface Design Solution for Bloated Software*. Technischer Bericht, University of Toronto and University of British Columbia, 2002.
- [37] MCNERNEY, T.: *Tangible Computation Blocks – Building-blocks for Physical Microworlds*. Technischer Bericht, Massachusetts Institute of Technology, 2001.
- [38] NIELSEN, J.: *Usability Engineering*. Morgan Kaufmann, 2000.

- [39] NIELSEN, J.: *Jakob Nielsen's Alert Box, October 10, 2005: R.I.P. WYSIWYG*. URL, www.useit.com/alertbox/wysiwyg.html, 13. Nov.: 23:53, 2006.
- [40] NORMAN, D.: *The Design of Everyday Things*. Basic Books, 1990.
- [41] NORMAN, D.: *The Invisible Computer – Why good Products can fail, the Personal Computer is so complex, and Information Appliances are the Solution*. M.I.T. Press, 1998.
- [42] PREECE, J., Y. ROGERS und H. SHARP: *Interaction Design – Beyond Human-Computer Interaction*. Wiley and Sons, 2002.
- [43] PREECE, J., Y. ROGERS, H. SHARP, D. BENYON, S. HOLLAND und T. CAREY: *Human-Computer Interaction*. Addison–Wesley, 1996.
- [44] PSIK, T.: *Designing Multimodal Interaction for Configurable Distributed Systems*. Doktorarbeit, University of Technology, Vienna, 2004.
- [45] RASKIN, J.: *Das intelligente Interface – Neue Ansätze für die Entwicklung interaktiver Benutzerschnittstellen*. Addison–Wesley, 2001.
- [46] RASKIN, J.: *The Humane Interface – New Directions for Designing Interactive Systems*. Addison–Wesley, 2005.
- [47] RESNICK, M., F. MARTIN, R. BERG, R. BOROVOY, V. COLELLA, K. KRAMER und B. SILVERMAN: *Digital Manipulatives – New Toys to think with*. Technischer Bericht, Massachusetts Institute of Technology, 1998.
- [48] RODDEN, T., A. CRABTREE, T. HEMMINGS, B. KOLEVA, J. HUMBLE, K.-P. AKESSON und P. HANSSON: *Configuring the Ubiquitous Home*. Technischer Bericht, University of Nottingham, 2004.
- [49] SHNEIDERMAN, B.: *Designing the User Interface – Strategies for effective Human-Computer Interaction*. Pearson, Addison–Wesley, 2. Aufl., 1992.
- [50] SHNEIDERMAN, B.: *Creating Creativity – User Interfaces for Supporting Innovation*. Technischer Bericht, University of Maryland, 1999.
- [51] SHNEIDERMAN, B.: *User Interface Design – Effektive Interaktion zwischen Mensch und Maschine, Leitfaden für intelligentes Schnittstellendesign, Was Programmierer und Designer über den Anwender wissen müssen*. mitp-Verlag, 3. Aufl., 2002.
- [52] SHNEIDERMAN, B.: *Leonardo's Laptop – Human needs and the new Computing Technologies*. M.I.T. Press, 2003.

- [53] SHNEIDERMAN, B. und C. PLAISANT: *Designing the User Interface – Strategies for effective Human-Computer Interaction*. Pearson, Addison–Wesley, 4 Aufl., 2005.
- [54] TANIN, E., A. LOTEM, I. HADDADIN, B. SHNEIDERMAN, C. PLAISANT und L. SLAUGHTER: *Facilitating Data Exploration with Query Previews – A Study of User Performance and Preference*. Technischer Bericht, University of Maryland, 1999.
- [55] TECHDIGEST: *Bildquelle*. URL, www.techdigest.tv/gadgets, 04. Jan.: 13:38, 2007.
- [56] TSE, E., S. GREENBERG und C. SHEN: *GSI DEMO – Multiuser Gesture / Speech Interaction over Digital Tables by Wrapping Single User Applications*. Technischer Bericht, University of Calgary, 2006.
- [57] TSE, E., S. GREENBERG und C. SHEN: *Motivating Multimodal Interaction Around Digital Tabletops*. Technischer Bericht, University of Calgary, 2006.
- [58] TSE, E., S. GREENBERG, C. SHEN und C. FORLINES: *Multimodal Multiplayer Tabletop Gaming*. Technischer Bericht, International Workshop on Pervasive Gaming Applications, 2006.
- [59] TU-WIEN: *Bildquelle*. URL, www.ims.tuwien.ac.at/research/mobile/ocar/images.php, 18. Dez.: 21:18, 2006.
- [60] ULMER, B., H. ISHII und R. JACOB: *Tangible Query Interfaces – Physically Constrained Tokens for Manipulating Database Queries*. Technischer Bericht, Massachusetts Institute of Technology, 2003.
- [61] VERBOT, F.: *Bildquelle*. URL, www.verbots.com/verbotse.php, 22. Sept.: 12:44, 2006.
- [62] VÄLKKYNEN, P., I. KORHONEN, J. PLOMP, T. TUOMISTO, L. CLUITMANS, A. HEIKKI und S. HEIKKI: *A User Interaction Paradigm for Physical Browsing and Near-Object Control based on Tags*. Technischer Bericht, VTT Information Technology/Electronics, Finland, 2003.
- [63] WIKIPEDIA: *Macromedia Director*. URL, <http://de.wikipedia.org/wiki/Macromedia>, 18. April: 15:26, 2007.