Aroo: Integration von realer und virtueller Dokumentenverwaltung

Thomas Seifried

DIPLOMARBEIT

eingereicht am
Fachhochschul-Diplomstudiengang

Digitale Medien

in Hagenberg

im Juni 2007

© Copyright 2007 Thomas Seifried

Alle Rechte vorbehalten

Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Hagenberg, am 13. November 2007

Thomas Seifried

Inhaltsverzeichnis

Erklärung iii

Vorwort vii

Kurzfassung viii

Abstract ix

1 Einleitung 1
- 1.1 Mythos des papierlosen Büros . . . 1
- 1.2 Organisation von Dokumenten . . . 2
- 1.3 Suche nach Dokumenten . . . 2

2 Konzept 4
- 2.1 Ausgangspunkt . . . 4
  - 2.1.1 Umgebung . . . 4
  - 2.1.2 Verwendung der Bürogegenstände . . . 4
  - 2.1.3 Organisation von Dokumenten . . . 6
  - 2.1.4 Arbeiten mit Dokumenten . . . 7
- 2.2 Probleme . . . 9
  - 2.2.1 Inkompatibilität . . . 9
  - 2.2.2 Zeitverlust . . . 9
  - 2.2.3 Materialverschwendung . . . 9
- 2.3 Szenario . . . 10
- 2.4 Ziele . . . 11
- 2.5 Anforderungen . . . 13
- 2.6 Ansatz . . . 14

3 Implementierung 16
- 3.1 Übersicht . . . 16
- 3.2 Technologieentscheidungen . . . 16
  - 3.2.1 Verbindung virtueller Ordner und realer Mappen . . . 16
  - 3.2.2 Direkte Ausgabe an den realen Mappen . . . 18
  - 3.2.3 Direkte Eingabe an den realen Mappen . . . 18

3.2.4 Darstellung und Interaktion mit virtuellen Daten . . . 19
3.2.5 Datenhaltung . . . 20
3.3 Setup . . . 21
3.4 Hardware Implementierung . . . 22
3.4.1 Überblick . . . 22
3.4.2 Hängemappenständer . . . 22
3.4.3 Hängemappe . . . 24
3.4.4 Tisch . . . 25
3.5 Software Implementation . . . 26
3.5.1 Überblick . . . 26
3.5.2 Technologien . . . 28
3.5.3 Mapping . . . 32
3.5.4 OneNote Add-in . . . 33
3.5.5 Tisch . . . 41

4 Resultate **49**
4.1 Navigation und Suche . . . 49
4.1.1 Bidirektionale Synchronisierung der Navigation . . . 49
4.1.2 Erweiterte Suche . . . 50
4.2 Besprechungen . . . 50
4.3 Notizen auf der Mappe . . . 53

5 Diskussion **54**
5.1 Ablegen von Dokumenten . . . 54
5.2 Darstellung virtueller Inhalte . . . 54
5.3 Besprechungsvorbereitung . . . 55
5.4 Interaktion am Tisch . . . 55
5.5 Suche . . . 55
5.6 Anwendungsspezifische Einschränkungen . . . 55
5.6.1 Komplexität virtueller DMS . . . 55
5.6.2 Falsche Einordnung . . . 56
5.6.3 Wartung . . . 56
5.7 Technologische Probleme . . . 56
5.7.1 OneNote . . . 56
5.7.2 DiamondTouch . . . 57
5.7.3 Tracking . . . 58
5.7.4 Digitale Stifte . . . 58
5.7.5 Widget Bibliothek . . . 58
5.7.6 VoodooIO . . . 58

6 Verwandte Arbeiten **59**
6.1 Augmentierung von Papier . . . 59
6.2 Speicherung über Platzhalter . . . 60
6.3 Integration von Papier in DMS . . . 60

6.4 Verknüpfung von virtuellen und realen Dokumenten 61
6.5 Pin&Play / VoodooIO 62

7 Resumé und Ausblick **63**

A Addin Quellcode **65**
A.1 AddinSample.cs 65

B Inhalt der DVD-ROM **69**
B.1 Diplomarbeit 69
B.2 Quellcode 69
B.3 Ausführbare Dateien 69
B.4 Sonstiges 69

Literaturverzeichnis **71**

Vorwort

Mein größter Dank gilt Michael Haller für die tatkräftige Betreuung dieser Diplomarbeit und der kontinuierlichen Hilfe in vielen Bereichen, ohne die diese Diplomarbeit nicht zustande gekommen wäre. Weiters gilt mein Dank Matt Jervis, der mir die VoodooIO Hardware zur Verfügung gestellt und die „smart folders“ entwickelt hat, und Jürgen Zauner, der mir bei der Softwareentwicklung eine große Hilfe war und auch die von mir verwendeten Bibliotheken JazzUp, AnotoPen und Widget entwickelt hat. Ich möchte auch Jakob Leitner, Daniel Leithinger und Peter Brandl danken, die mich beim Spinnen des Konzepts und der Entwicklung dieser Arbeit immer wieder auf neue Ideen gebracht haben und auch wie Frieda Raschhofer, Christine Seifried und Karin Hansbauer mir beim Korrekturlesen eine große Hilfe waren. Nicht zuletzt gilt mein Dank meiner Freundin Karin und meiner Familie für die moralische Unterstützung während des Studiums und des Schreibens dieser Arbeit.

Kurzfassung

Seit Jahren wird das papierlose Büro von vielen Seiten vorhergesagt. Der Computer sollte das Papier komplett ersetzen und die Arbeit mit Dokumenten nur mehr digital geschehen. Trotzdem ist das papierlose Büro heute noch immer ein Mythos. Nach wie vor bevorzugen viele Leute das Lesen von Text auf Papier anstatt auf dem Bildschirm. Somit werden Dokumente im Büro üblicherweise doppelt abgelegt: zum Einen digital am Computer und zum Anderen als Ausdruck. Aber anders als am Computer kann das Auffinden von Dokumenten in einem physikalischen Ordnersystem sehr zeitaufwendig werden. Das Problem liegt hierbei zum einen an der fehlenden Verbindung zwischen der realen und virtuellen Version von Dokumenten und zum anderen an der fehlenden Ein- und Ausgabemöglichkeit am Ordnersystem selbst.

Aroo stellt eine Kombination aus virtuellen und physikalischen Benutzerschnittstellen auf einem erweiterten Ordnersystem zur Verfügung. Es verbindet die virtuelle und reale Dokumentenorganisation und erweitert die Suche in traditionellen Ordnersystemen. Auf Mappen und Registraturen angebrachte Geräte ermöglichen eine direkte Interaktion mit dem Computer. Zusätzlich ermöglicht die Erweiterung eines Besprechungstisches um ein interaktives Tabletopdisplay die nahtlose Integration von virtuellen und realen Dokumenten.

Mit all diesen Erweiterungen wurde die Arbeit mit virtuellen und realen Dokumenten und deren Organisation signifikant verbessert. Eine formale Benutzerstudie ist zwar noch ausständig, doch die informalen Rückmeldungen von ersten Testbenutzern sind größtenteils positiv ausgefallen.

Abstract

Although over the last few decades, the future paperless office has been predicted and promised by many people, it still remains a myth. Most people still prefer to review their documents in printed form. In most work environments people archive both the real and digital version of their documents. But unlike the digital world, in the physical world locating a document can become a time consuming task. The problem is often related to the fact that there is no direct connection between the physical and digital representation of documents, even when there is an exact copy of the physical document on the computer.

The *Aroo* system provides a combination of virtual and physical interface to an augmented filing cabinet system. It links the virtual and physical document storage and retrieval capabilities of the traditional filing cabinets and folders to that of a computer system. Folders and cabinets are extended with devices for direct input and output. Furthermore, an interactive table and digital pens enable a seamless interaction with virtual as well as physical documents.

With all those extensions the work with both virtual and physical document management systems improves significantly. Although a formal user-study needs to be conducted the informal results seem promising.

Kapitel 1

Einleitung

Büroarbeit wird seit jeher von Akten und Dokumenten begleitet. In vielen Bereichen bilden sie sogar das Zentrum der Arbeit selbst. Die Verwendung von ihnen ist mannigfaltig und sie werden zu allen möglichen Arbeitsschritten eingesetzt. Sie werden geschrieben, gelesen, geändert, verteilt, besprochen und zuletzt auch abgelegt und später wieder gesucht.

1.1 Mythos des papierlosen Büros

Seit Jahren wird das papierlose Büro von vielen Seiten vorhergesagt. Der Computer sollte das Papier komplett ersetzen und die Arbeit mit Dokumenten nur mehr digital geschehen. Somit sollte der Medienbruch überwunden und Material und Zeit gespart werden. Trotzdem ist das papierlose Büro heute noch immer ein Mythos [34]. Nicht nur, dass Papier jeden Tag verwendet wird, sondern der Gebrauch ist in den letzten Jahren sogar noch gestiegen. Das liegt unter anderem daran, dass viele Leute das Lesen von Text am Papier bevorzugen. Papier hat, obwohl es ein sehr altes Medium ist, noch immer viele Vorteile gegenüber einem Computerbildschirm. Ausdrucke auf Papier sind leicht und flexibel, relativ robust, einfach zu kommentieren, einfach zu verwenden, haben eine sehr hohe Auflösung, benötigen keine Hintergrundbeleuchtung und funktionieren auch ohne Strom [15, 34].

Papier hat aber auch Nachteile gegenüber dem Computer und dem Bildschirm. Das Hauptproblem ist die fehlende Interaktivität. Papier ist ein „Dead End“ [17]. Ist ein Dokument einmal ausgedruckt, so ist die Verknüpfung zum digitalen Inhalt verloren. Kommentare oder Änderungen des Textes am Papier müssen bei Bedarf zusätzlich in den Computer eingegeben werden. Doch da diese Tätigkeit mit einem realativ hohen Zeitaufwand verbunden ist, werden diese oft auch einfach am Papier belassen und so meistens vergessen. Zusätzlich ist das Auffinden der digitalen Version eines Ausdrucks oft nicht einfach. Ohne entsprechende Hilfsmittel (z.B. eine Kennnummer auf dem Ausdruck) ist hier der Zeitaufwand hoch.

1.2 Organisation von Dokumenten

Dokumente werden im Büro üblicherweise doppelt abgelegt: zum einen digital am Computer um zum anderen als Ausdruck. Ausdrucke werden dabei auf dem Arbeitsplatz geordnet und gestapelt oder in einer besser organisierten Form in einem Ordnersystem abgelegt. Verschiedenste Arten solcher Organisationssysteme wie Aktenordnerschränke, Registraturen für Schnellhefter und Hängeregistraturen wurden seit Mitte des 19. Jahrhunderts entwickelt. Letztere, die Hängemappe, wurde 1898 von Edwin G. Seibels erfunden [37]. Diese über 100 Jahre alte Erfindung wurde seit damals nur geringfügig verändert. Nur die verwendeten Materialien zur Herstellung der Hängemappen und Registraturen und deren Ausmaße wurden modifiziert. Das Grundprinzip blieb seither dasselbe und Papier wird noch immer genauso organisiert wie seit der Erfindung dieser Ordnersysteme.

Die Erfindung des Computers revolutionierte die Arbeit mit Dokumenten und ermöglichte deren digitale Organisation unter anderem in so genannten *Document Management Systems (DMS)*. Damit haben sich die technischen Möglichkeiten zwar stark verändert, doch das Ordnungsschema am Computer ist dasselbe wie jenes von realen Ordnersystemen geblieben. Während sich reale Dokumente am Arbeitstisch stapeln, häufen sich Dateien am virtuellen Desktop und sowohl in der realen als auch in der virtuellen Welt werden Dokumente in Ordnern abgelegt.

Wie aber schon beim Papier besteht hier keine Verbindung zwischen realen und virtuellen Ordnern und somit werden sie zweimal abgelegt, zum einen am Computer und zum anderen in der Registratur oder im Regal.

1.3 Suche nach Dokumenten

Die Trennung zwischen virtueller und physikalischer Organisation von Dokumenten erweist sich besonders beim Suchen nach Dokumenten als ein Problem. Ein Suchvorgang am Computer ist relativ einfach und schnell. Über die Eingabe einiger Suchbegriffe erhält man eine Liste von gefunden Dokumenten. Diese kann rasch manuell durchsucht und gewünschte Dokumente geöffnet und gelesen werden. Dagegen kann eine Suche in einem realen Ordnersystem enorm zeitaufwendig werden. Entspricht die Suche nicht dem Ordnungsschema (z.B. Suche nach Dokumenten von einem bestimmten Datum in einem nach Kunden sortierten System), so müssen alle Ordner durchsucht werden, damit jeder gesuchte Ausdruck gefunden werden kann. Selbst wenn die gesuchten Dokumente ebenfalls am Computer verfügbar sind, hilft die Suche am Computer nicht beim Auffinden der Ausdrucke, wenn eine Verbindung zwischen virtueller und physikalischer Organisation fehlt.

Weiters können reale Ordnersysteme nur reale Objekte fassen. Am Computer können neben Textdokumenten auch Videos, Töne, 3D-Modelle Ani-

mationen und andere dynamische oder interaktive Inhalte gespeichert, gesucht und angezeigt werden. Textdokumente und Grafiken können auf Papier zwar sehr gut abgebildet werden, doch rein virtuelle Inhalte können nicht oder nur schwer mit Papier in eine physikalische Form gebracht werden. Es besteht zwar die Möglichkeit, diese Inhalte auf einem Datenträger gespeichert abzulegen, allerdings ist der Inhalt auf diesen für uns nicht sichtbar.

Das Problem liegt in erster Linie dabei, dass reale Ordner und Mappen keine Möglichkeit zur direkten Ein- und Ausgabe besitzen. Sie können weder virtuelle Inhalte anzeigen, noch unterstützen sie das Anzeigen einer „Ergebnisliste“ beim Suchen.

Kapitel 2

Konzept

2.1 Ausgangspunkt

Dokumente gibt es in verschiedensten Formen und Größen und werden je nach Gebrauch an verschiedenen Orten und mit verschiedensten Hilfsmitteln erstellt, bearbeitet, gelesen, abgelegt oder anderweitig verwendet. Je nach Zweck des Dokuments und der Umgebung sind diese in der Form, in deren Bearbeitungsmitteln und deren Ablagemöglichkeiten angepasst. Zum Beispiel wird ein Strafzettel im Freien erstellt, braucht relativ wenig Informationen zu beinhalten und muss aber auch im Freien abgelegt werden können. Daraus ergibt sich, dass ein kleines Blatt Papier und ein wasserfester Kugelschreiber die passende Kombination sind. Beispielsweise bei Bauplänen ist die Situation aber völlig anders. Die Form, das Bearbeitungsmittel und der Umgang unterscheiden sich stark vom Beispiel der Strafzetteln.

Eine elektronische Unterstützung für die Organisation von Dokumenten hängt somit stark von dem Verwendungszweck von Dokumenten, aber auch von der Umgebung in der sie verwendet werden, ab.

2.1.1 Umgebung

Für *Aroo* wurde das Büro als Umgebung ausgewählt. In der täglichen Arbeit werden die Bürobediensteten mit Dokumenten unterschiedlichsten Zwecks und Inhalts konfrontiert. Diese Dokumente spielen hier bei vielen Tätigkeiten eine zentrale Rolle.

Wie in Abbildung 2.1 dargestellt, besteht das für dieses Konzept gewählte Büro aus drei *Hauptbestandteilen*: einem Computerarbeitsplatz, einem Ordnersystem und einem Besprechungstisch.

2.1.2 Verwendung der Bürogegenstände

Jeder der beschriebenen Gegenstände erfüllt im Büro einen bestimmten Zweck bei der Arbeit mit Dokumenten.

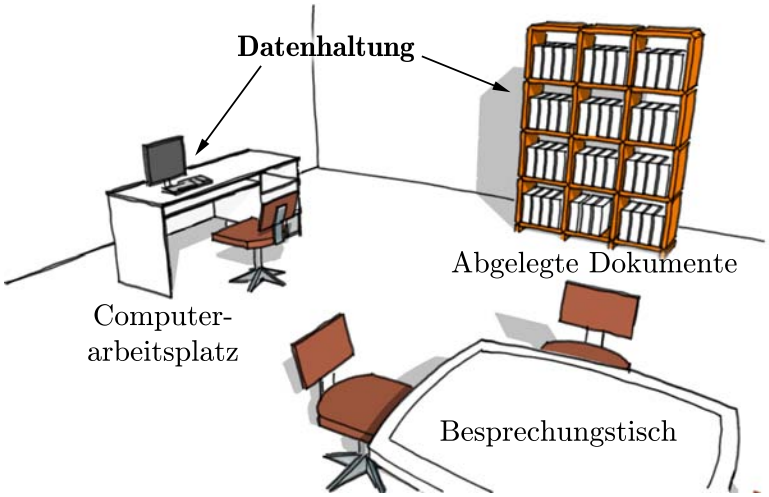

Abbildung 2.1: Ein einfaches Büro mit den drei wichtigsten Bestandteilen: Der Computerarbeitsplatz, ein Hängeregister oder Regal und ein Besprechungstisch. Die Datenhaltung erfolgt sowohl elektronisch im Computer als auch physikalisch im Hängeregister.

Am Computerarbeitsplatz wird die meiste Zeit verbracht. Ein Großteil der Arbeit beschäftigt sich mit der elektronischen Bearbeitung von Dokumenten. Das reicht von der Kommunikation über E-Mail bis zum Verfassen von Briefen und anderen Schriftstücken sowie auch der gesamten Organisation von Dokumenten. Doch an einem Arbeitsplatz wird nicht nur am Computer gearbeitet, sondern auch am Tisch selbst. Ausdrucke werden am Tisch abgelegt, gestapelt, gelesen und bearbeitet. Während an anderen Orten oft auch zu zweit oder in einer Gruppe gearbeitet wird, wird dieser Platz meistens nur von einer Person gleichzeitig benutzt.

Der zweite wichtige Gegenstand im Büro ist das Ordnersystem. Dieser ist neben dem Arbeitstisch der zweite Ort, an dem Dokumente abgelegt werden. Im Gegensatz zum Arbeitstisch geschieht dies hier aber nach einem geordnetem Schema und ist für eine größere Menge und längere Lagerdauer gedacht. Ausdrucke werden nach bestimmten Gesichtspunkten sortiert in Mappen oder Ordner abgelegt und in Regale oder Hängeregistraturen gegeben. Dies geschieht meist parallel zu der Organisation von Dokumenten am Computer.

Der Besprechungstisch ist im Büro jener Ort, an dem üblicherweise mehr als eine Person gleichzeitig arbeitet. Bei Besprechungen wechselt daher das Arbeitsgeschehen vom Arbeitsplatz zum Besprechungstisch. Dokumente werden an diesem Ort praktisch nur in der physikalischen Form verwendet, da sie in dieser sehr angenehm zu lesen sind und auch leicht transportiert werden können [15,34]. Auch werden diese Dokumente gleichzeitig von meh-

reren Personen verwendet.

2.1.3 Organisation von Dokumenten

Wie schon erwähnt, werden Dokumente in Büros meistens doppelt abgelegt, nämlich einerseits digital am Computer andererseits analog in ausgedruckter Form in Mappen oder Ordnern. Die Organisation von Dokumenten weist in beiden Welten Ähnlichkeiten auf. Es wird oft die selbe Ordnungsstruktur verwendet und auch die Art und Weise der Organisation am Computer mittels Ordnern und Dateien ähnelt der von realen Ordnersystemen. Ordner werden nach bestimmten Kriterien sortiert (z.B. alphabetisch oder chronologisch) und in Regale oder Register gegeben, in denen Zusammengehörendes abgelegt wird. Dokumente werden nach weiteren Kriterien in diesen Ordnern abgelegt. Die Kriterien sind dabei vom Verwendungszweck abhängig. Beim Erstellen einer wissenschaftlichen Arbeit könnten die benötigten Dokumente je nach Thema sortiert werden, beim Archivieren des Briefwechsels beispielsweise nach Kunden. Diese hierarchische Organisation wird dabei sowohl in der physikalischen als auch der virtuellen Welt verwendet. Da die Kriterien für die Organisation am Computer meistens dieselben sind, werden Dokumente am Computer in den selben Gruppen zusammengefasst wie auch im realen Ordnersystem.

Abbildung der Realität

Das elementare Organisationssystem am Computer ist das Dateisystem und ist realen Ablagesystemen nachgebildet (siehe Abb. 2.2). Dokumente werden aus Text, Zeichnungen und Diagrammen erstellt und dienen somit als Container für diese Inhalte. Im weiteren werden diese Dokumente in passende Ordner abgelegt. Diese Ordner beinhalten Objekte, die einen bestimmten Bezug zueinander haben. Weiters gibt es in den meisten Desktopcomputern auch einen so genannten virtuellen Arbeitsplatz, auf dem Dokumente und Ordner abgelegt werden können [25, 35]. Die Idee dabei ist, dort alle Daten abzulegen, die häufig verwendet werden und somit schnell erreichbar sein sollten. Graphische Benutzerschnittstellen (GUI) unterstützen diese Metaphern zusätzlich durch die Verwendung bekannter Begriffe und Formen. Ordner werden üblicherweise als Aktenmappe und Dokumente als ein Ausdruck auf Papier dargestellt und am virtuellen Arbeitsplatz lassen sich diese, wie auch am realen Arbeitstisch, beliebig anordnen [1].

Die Benutzung des Computers sollte durch diesen Bezug zur Wirklichkeit erleichtert werden, wobei die Möglichkeiten im Bereich der Dokumentenverwaltung am Computer natürlich um einiges größer sind. Anders als in der Realität können Hierarchien beinahe beliebig tief sein und auch Querverweise sind möglich.

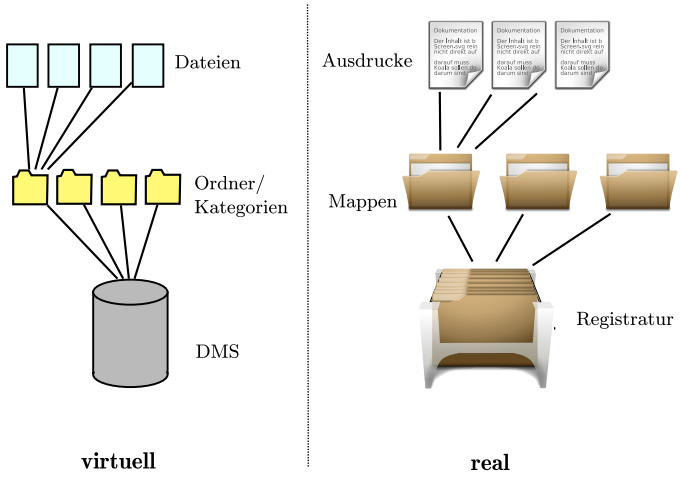

Abbildung 2.2: Die Datenhaltung am Computer gegenüber jener im Hängeregister. Am Computer wird versucht, reale Ordnersysteme nachzubilden.

Virtueller vs. realer Organisation

Durch die elektronische Datenverarbeitung wurden neue Möglichkeiten zur Darstellung und Bearbeitung von Daten und Dokumenten geschaffen. Neben statischen Dokumenten wie Texte, Bilder, Diagramme oder Zeichnungen können auch dynamische Inhalte wie Audio, Video, Animationen und Multimediapräsentationen erstellt, betrachtet und bearbeitet werden. Physikalische Dokumente hingegen können nur auf statischen Materialen wie z.B. Papier „aufgetragen“ und auf diesen betrachtet werden. Änderungen sind nur mit großem Aufwand annullierbar und die Darstellung von dynamischen Inhalten (z.B.: Audio, Video, Animationen) ist gar nicht möglich.

In Folge dessen gibt es nicht für jede Art von Dokument ein physikalisches Gegenstück. Die Tabelle 2.1 gibt einen kurzen Überblick über virtuelle und reale Repräsentationen von Objekten in einem DMS.

2.1.4 Arbeiten mit Dokumenten

Ein weiterer Aspekt ist das Arbeiten mit Dokumenten. Im Büro ist ein beträchtlicher Anteil an Tätigkeiten mit der Verwendung von Dokumenten verbunden. Dabei durchlaufen sie eine Reihe von Arbeitsschritten, die an den verschiedenen Orten des Büros stattfinden. In der Abbildung 2.3 ist dargestellt, wie ein solcher „Dokumentenfluss“ aussehen kann.

Zunächst wird das Dokument erzeugt. Dies kann durch die Erstellung am Computer oder auch durch das Beziehen aus einer externen Quelle (z.B.: In-

virtuell	**real**
Hauptordner	Regal
Ordner	Mappe
Dokument	Ausdruck
Textdokument	Textausdruck
Plan	Planausdruck
Bild	Fotoabzug
Ton	n/a
Video	n/a
3D Modell	n/a
Interaktive Animation	n/a

Tabelle 2.1: Überblick über virtuelle und reale Repräsentationen von Objekten in einem DMS.

ternet, E-Mail, usw.) geschehen. Am Computer wird das Dokument dann weiter bearbeitet und abgelegt. In einem weiteren Schritt werden viele von diesen ausgedruckt, da Text auf Papier angenehmer zu lesen und in Besprechungen leichter zu verwenden ist als ein Computer [15, 34]. Nach der Benutzung landen diese Dokumente in der Regel im Müll oder, wenn sie als wichtig erachtet werden, in einem Aktenordner.

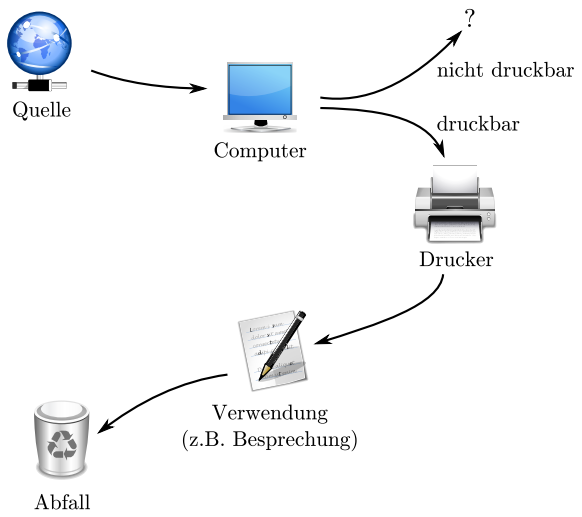

Abbildung 2.3: Dokumente durchlaufen während der Verwendung im Büro verschiedene Stationen.

2.2 Probleme

Da zwischen virtueller und realer Repräsentation von Dokumenten keine computergestützte Verbindung besteht, entstehen folgende Probleme:

- Inkompatibilität
- Zeitverlust (beim Suchen und Ablegen)
- Materialverschwendung

Diese lassen sich am besten auf Grund des „Dokumentenflusses“ (wie in Abschnitt 2.1.4 beschrieben) im Büro beschreiben.

2.2.1 Inkompatibilität

Das erste Problem ist die Druckbarkeit von Dateien. Wie auch in Abbildung 2.8 gezeigt wird, gibt es nicht für alle Dokumente eine physikalische Repräsentation oder ist diese für den gewünschten Zweck nicht sinnvoll. Zum Beispiel ist der Ausdruck von Videos nicht möglich und die meisten Datenträger von Videos haben keine Möglichkeit, deren Inhalt anzuzeigen.

Vor allem bei Besprechungen kann es aber notwendig sein, genau diese Dokumente am Ort der Besprechung (z.B. ein Besprechungstisch) betrachten zu können, da es ansonsten zu einer Unterbrechung durch den Ortswechsel zum Computer kommt [33]. Folglich wird die Teamarbeit gestört, weil zum einen nur eine Person gleichzeitig den Computer steuern kann und zum anderen die Kommunikation in der Gruppe behindert wird. [10].

2.2.2 Zeitverlust

Ein zweites Problem ist der Zeitverlust beim Suchen von bereits ausgedruckten Dokumenten im Ordnersystem. Im Gegensatz zum Computer gibt es keine Suchfunktion, die dem Benutzer erlaubt mit beliebigen Suchbegriffen die Dokumente im Register zu durchsuchen. Die einzige Hilfe, die angeboten wird, ist die hierarchische Ordnung und eine gewisse Sortierung der Mappen. Somit kann das Suchen nach Dokumenten sehr zeitaufwendig werden, vor allem dann, wenn die gesuchten Dokumente nicht dem Ordnungsschema entsprechen, in der die Mappen gefüllt und sortiert worden sind. Dieselben Schwierigkeiten treten beim Ablegen von Dokumenten auf. Aus diesem Grund werden solche Register auch oft als „Dead End“ bezeichnet [17]. Alle Dokumente, die einmal abgelegt worden sind, werden nie mehr oder nur selten wieder verwendet, da der Zeitaufwand für die Suche zu groß ist.

2.2.3 Materialverschwendung

Daraus folgt, dass viele Dokumente bei Bedarf nochmals ausgedruckt werden. Die Suche am Computer ist viel schneller erledigt und auch der Aus-

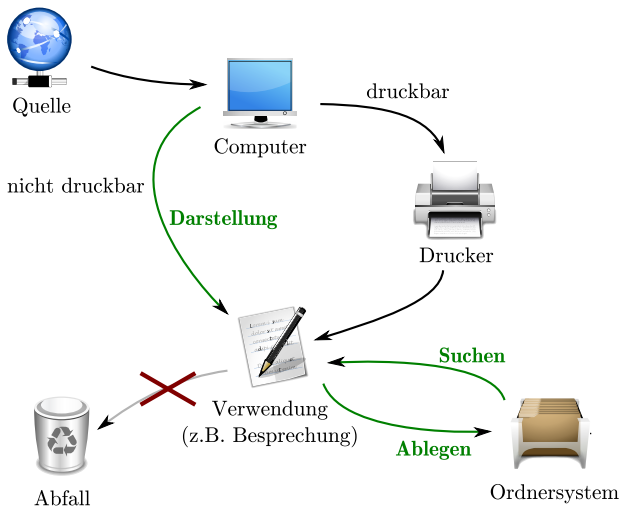

Abbildung 2.4: Einige Probleme beruhen auf der fehlenden Unterstützung von der Suche und dem Ablegen von Ausdrucken und der Darstellung von nicht druckbaren Dokumenten. Ausdrucke sollen nicht nur einmal verwendet werden, sondern nach der Verwendung später wieder benutzt werden können.

druck dauert bei kurzseitigen Dokumenten nur wenige Minuten. Die Ausdrucke landen dann nach der Verwendung meistens sofort wieder im Müll.

2.3 Szenario

Wie beschrieben wurde, ist die Organisation von virtuellen (elektronische Dateien) und realen Formen (bedrucktes Papier) von Dokumenten in heutigen Büros größtenteils voneinander getrennt und es gibt kaum Verbindungen zwischen diesen. Wie in der Zukunft die Arbeit in einem Büro aussehen könnte, in dem diese Verbindung besteht, soll anhand des folgenden Szenarios gezeigt werden.

Dieses konkrete Szenario spielt sich im Kontext des Verfassens einer wissenschaftlichen Arbeit ab. Eine Autorin soll über ihr Projekt einen Konferenzbeitrag verfassen und wird dabei von ihrem Betreuer unterstützt. Sie treffen sich zweimal in der Woche zu einem gemeinsamen Treffen, um die Inhalte und den aktuellen Fortschritt der Arbeit zu diskutieren. Die Autorin hat bereits viele verwandte und für das Verfassen notwendige Dokumente gesammelt. Diese Unterlagen werden primär am Computer verwaltet und Ausdrucke in einem realen Ordnersystem abgelegt.

Das Szenario beschreibt eine Besprechungssituation, bei der die Autorin der zu schreibenden Arbeit eine kurze Unterredung mit ihrem Betreuer

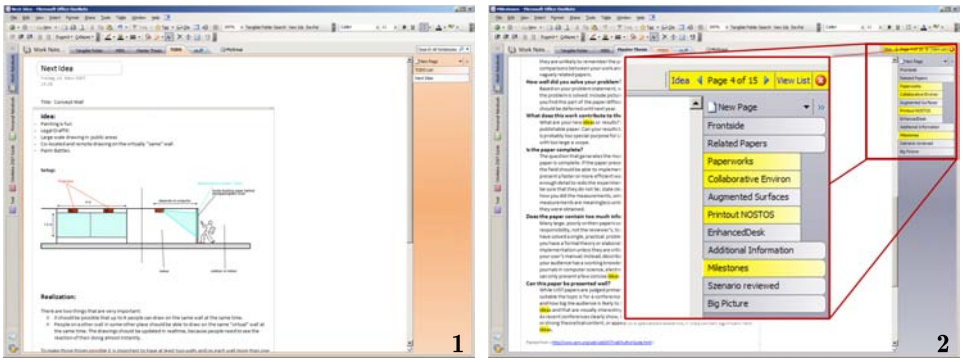

Abbildung 2.5: (1) Alle für die Arbeit relevanten Dokumente sind am Rechner gespeichert. (2) Die Autorin sucht nach einem bestimmten Begriff, und die Ergebnisse werden am Bildschirm angezeigt.

hat. Dafür benötigt sie die Ausdrucke von einigen Unterlagen, die bereits im System abgelegt worden sind. Eine Suchanfrage am Computer sollte das Suchen nach diesen Unterlagen beschleunigen (Abb. 2.5). Es werden sofort auch einige Dokumente gefunden und am Computerbildschirm angezeigt. Im selben Augenblick werden auch die zugehörigen realen Ordner im Regal optisch hervorgehoben (Abb. 2.6). Die Autorin geht zum Regal und durchsucht nur mehr jene Ordner, die durch die Suchanfrage hervorgehoben wurden und nimmt alle notwendigen heraus, um sie für die Besprechung griffbereit zu haben.

Die Besprechung findet am Besprechungstisch des Büros statt. Beide Personen diskutieren über die Arbeit und verwenden dazu die ausgedruckten Unterlagen, die zuvor ausgewählt worden sind. Während der Besprechung wollen die beiden weiterführendes Videomaterial zu einer Arbeit sehen. Um das zu erreichen, legt der Betreuer den Ordner, der mit diesem Material am Computer verknüpft ist, auf den Tisch. Am Tisch wird der reale Ordner erkannt und es werden alle nicht ausgedruckten Inhalte des verknüpften virtuellen Ordners auf den Tisch dargestellt (Abb. 2.7). Die beiden Besprechungsteilnehmer können nun sowohl mit den ausgedruckten als auch dem Video arbeiten, ohne den Tisch zu verlassen.

2.4 Ziele

Das Hauptziel der Arbeit ist es, die in 2.2 beschrieben Probleme der *Inkompatibilität*, des *Zeitverlusts* und der *Materialverschwendung* zu lösen oder zumindest zu reduzieren.

Dass bei Besprechungen undruckbare Dokumente nicht verwendet werden können, führt oft zu einer Störung der Besprechung. Da ein Besprechung meist an einem Tisch stattfindet, soll eine Möglichkeit gefunden werden,

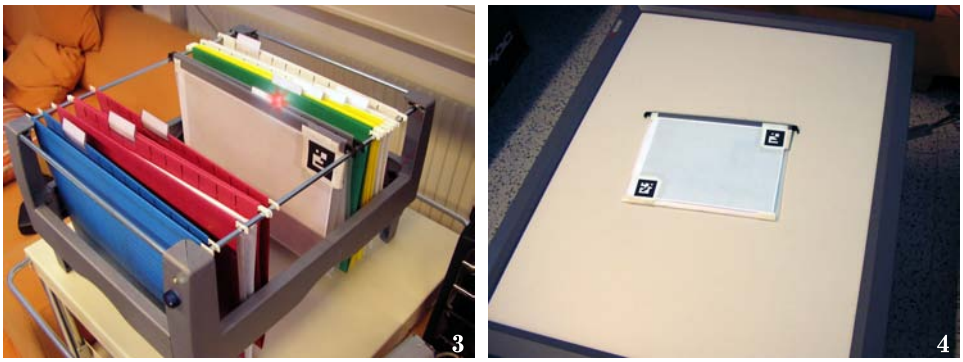

Abbildung 2.6: (3) Die Suchergebnisse werden auch an den realen Ordnern angezeigt, indem ein LED zu leuchten beginnt. (4) Der Ordner wird auf den Tisch gelegt und er wird von der Anwendung erkannt.

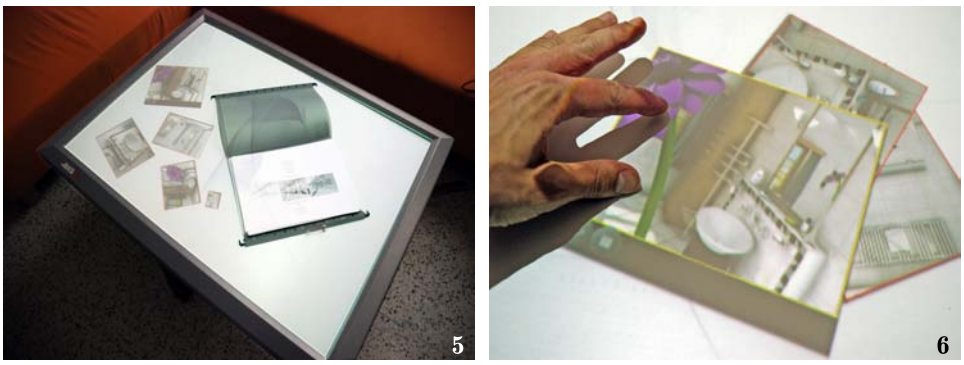

Abbildung 2.7: (5) Es werden alle Inhalte des Ordners auf den Tisch projiziert, die nicht schon im Ordner selbst vorhanden sind. (6) Nach dem Öffnen des Ordners sind die Objekte am Tisch und sie können in der Besprechung benützt werden.

undruckbare Dokumente zumindest am Tisch darzustellen. Weiter soll auch eine intuitive Mehrbenutzerinteraktion, angelehnt an den Umgang mit realen Ausdrucken auf Papier, möglich sein.

Zusätzlich soll das Problem des Zeitverlusts und der daraus folgenden Materialverschwendung verbessert werden. Das Suchen und Ablegen von Dokumenten in ein Hängeregister soll durch Computerunterstützung beschleunigt werden.

Grundsätzlich soll die Lücke zwischen der virtuellen und realen Organisation von Dokumenten und somit auch die zwischen virtuellen und realen Repräsentationen von Dokumenten zu verkleinert werden. Wie in Abbildung 2.8 dargestellt ist, soll eine Verbindung zwischen einem Document Management System und einem Ordnersystem und virtueller Ordner und realer

Mappen erstellt werden.

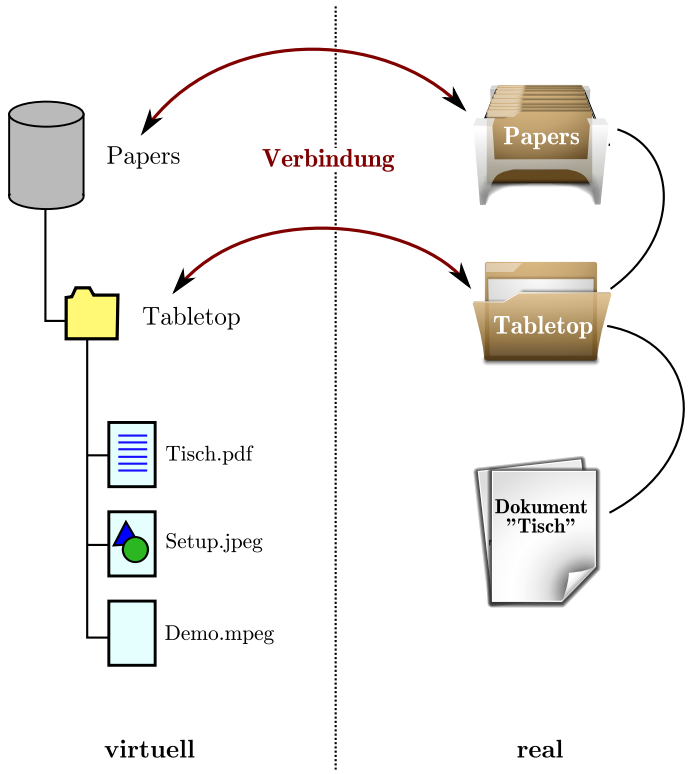

Abbildung 2.8: Das Ziel ist eine Verbindung zwischen realen und virtuellen DMS durch das Verknüpfen von virtuellen Ordnern und realen Mappen zu erzeugen. Dabei ist zu beachten, dass nicht alle virtuellen Dokumente auch in einer physikalischen Form verfügbar sind.

2.5 Anforderungen

Aus den gesetzten Zielen ergeben sich folgende Anforderungen für das zu entwickelnde System:

- Die Herstellung einer Verbindung zwischen den virtuellen Ordnern und den realen Mappen sowie zwischen dem virtuellen DMS und dem realen Ordnersystem.
- Das reale Ordnersystem benötigt eine direkte Ausgabemöglichkeit, um die Suche nach Mappen oder Ordnern zu beschleunigen und die Brücke vom virtuellen auf das reale System zu schlagen.
- Das reale Ordnersystem benötigt eine direkte Eingabemöglichkeit, damit auch direkt vom realen System aus Eingaben getätigt werden

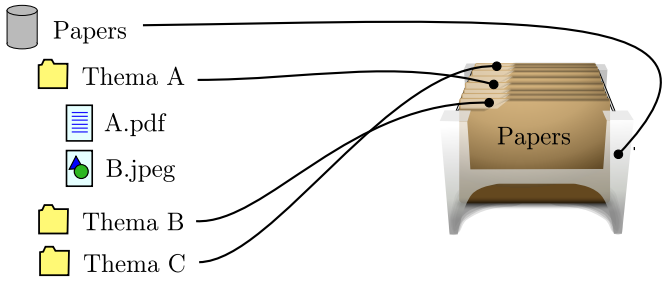

Abbildung 2.9: Die Verknüpfung erfolgt auf Ebene der Registraturen und Mappen. Die Reihenfolge der Mappen in der Registratur muss dabei nicht die selbe wie im virtuellen System sein.

können und somit eine Brücke vom realen auf das virtuelle System geschlagen werden kann.

- Neben der Suche im DMS am Computer muss auch eine computerunterstützte Suche im realen Ordnersystem möglich sein.
- Rein virtuelle Daten müssen in einer für Gruppenarbeit tauglichen Form dargestellt werden.
- Eine Einbindung in ein bestehendes System ist zu erstellen, um die zu entwickelnde Anwendung näher an eine reale Situation zu bringen.

2.6 Ansatz

Um die Lücke zwischen virtuellen und realen Dokumenten zu verkleinern, müssen die drei beschriebenen Orte des Büros, Arbeitsplatz, Ordnersystem und Besprechungstisch, erweitert werden.

Am Computer des Arbeitsplatzes soll das bereits verwendete DMS um die erweiterte Suchfunktion und einer bidirektionalen Navigation angepasst werden. Die Suchergebnisse sollen nicht nur am Computerbildschirm, sondern auch am Ordnersystem direkt angezeigt werden. Auch die Navigation durch Ordner am Computer soll am Ordnersystem ebenfalls visualisiert werden. Die Sortierung der realen Ordner kann dabei völlig anders sein als jene im Computer (siehe Abb. 2.9). Da Suchergebnisse und aktuell geöffnete Ordner so schneller erkennbar sind, soll die Suche nach Ausdrucken und das Ablegen von diesen in bestimmte reale Ordner beschleunigt werden. Damit das funktionieren kann, muss einerseits eine Verbindung zwischen virtuellen und realen Ordnern erstellt werden und andererseits die Inhalte dieser im virtuellen wie im realen System die selben sein. Zusätzlich muss jeder reale und virtuelle Ordner eindeutig identifizierbar sein.

Das Ordnersystem soll um Ein- und Ausgabemöglichkeiten erweitert werden. Jeder Ordner oder jedes Regal erhält ein Ausgabegerät, das Suchergebnisse und die aktuelle Navigation am Computer anzeigen kann. Zusätzlich sollen auch direkt über die Ordner Eingaben getätigt werden können, damit auch der umgekehrte Weg vom realen ins virtuelle System möglich ist.

Reale Ordner können nun nicht nur reale Dokumente enthalten sondern auch mit virtuellen Dokumenten verknüpft sein. Um diese auch in einer Besprechungssituation betrachten zu können, muss der Tisch um ein Display erweitert werden. Legt man einen Ordner auf den Tisch, sollen die virtuellen Inhalte des Ordners direkt am Tisch angezeigt werden. Die Interaktion mit diesen virtuellen Objekten soll ähnlich einfach sein wie mit den Ausdrucken. Dafür muss am Tisch neben dem Display noch ein einfach zu bedienendes, mehrbenutzerfähiges Eingabegerät zur Verfügung stehen. Auch müssen die verschiedenen Ordner auf dem Tisch vom Computer mit geeigneten Mitteln identifiziert werden können.

Kapitel 3

Implementierung

3.1 Übersicht

In Abbildung 3.1 wird schematisch dargestellt, welche Technologien für welche Anforderungen eingesetzt werden. Aufgrund der Verwendung vieler verschiedenen Technologien ergibt sich bei der praktischen Umsetzung von *Aroo* ein Schwerpunkt auf die Zusammenführung dieser zu einer stimmigen Gesamtanwendung.

Das folgende Kapitel soll erklären, warum welche Technologie verwendet worden ist, welche Funktion diese im Gesamtprojekt haben und wie diese zusammengeführt worden sind.

3.2 Technologieentscheidungen

Die im Folgenden beschriebene praktische Umsetzung bedient sich einer Vielzahl von Technologien.

3.2.1 Verbindung virtueller Ordner und realer Mappen

Die Mappen werden laut Szenario (Abschnitt 2.3) hauptsächlich an zwei Orten verwendet: Im Hängemappenständer und am Besprechungstisch. Damit eine Verbindung zu den virtuellen Ordnern hergestellt werden kann, muss jede Mappe an beiden Orten eindeutig identifizierbar sein.

Verbindung am Ständer

Im Hängemappenständer sind die Mappen abgelegt und werden selten bewegt. Der Abstand untereinander ist gering und nur die Karteireiter sind gut zu erkennen. Die Mappen sollen im Hängemappenständer vom Arbeitsplatz aus leicht erreichbar sein. Ein Verbau (z.B. eine geschlossene Box) ist daher zu vermeiden.

In diesem Fall gibt es drei Möglichkeiten zur Identifizierung:

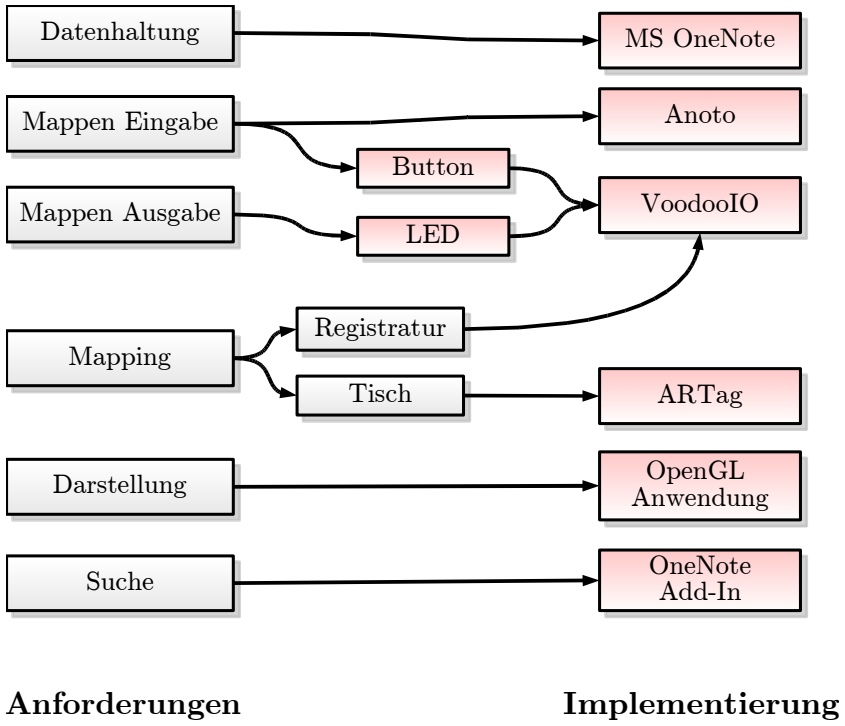

Abbildung 3.1: Mit den ausgewählten Technologien und Implementierungen sollen die Anforderungen erfüllt werden.

- optische Erkennung,
- drahtlose Erkennung oder
- elektronische, kontaktgebundene Erkennung.

Die Verwendung von Kameras zur optischen Erkennung der Mappen ist aufgrund der hohen Anzahl von Mappen und der damit leicht vorkommenden Verdeckung von markanten optischen Merkmalen der Mappen problematisch.

Die Verwendung von elektromagnetischen Wellen zur Identifizierung (z.B. Radio Frequency Identification (RFID)) wäre in Hinsicht der hohen Dichte an Mappen im Ständer durchaus möglich [41], doch ist die genaue Positionsbestimmung technisch schwierig realisierbar [30, 30].

In Blickpunkt auf die Anforderung einer direkten Ein- und Ausgabemöglichkeit an einer Mappe wurde eine rein elektronische und kontaktbasierte Realisierung gewählt. Aufgesetzt wurde dabei auf das Projekt VoodooIO [40] der *Embedded Interactive Systems Group*[1] der *Lancaster University*.

[1] http://eis.comp.lancs.ac.uk/

VoodooIO arbeitet mit einer Zwei-Leiter-Technik und benötigt daher nur zwei Kontakte um kommunizieren zu können (siehe Abschnitt 6.5). Das System wurde daher so konstruiert, dass über die beiden Hacken einer Hängemappe immer Kontakt zu den zwei Schienen im Ständer besteht. Über den VoodooIO-Controller ist der Ständer dann mit dem Computer verbunden.

Verbindung am Tisch

Im Gegensatz zum Ständer werden die Hängemappen auf dem Tisch frei bewegt. Auch sind immer nur einige wenige Mappen gleichzeitig dort und sie sind von oben gut erkennbar. Da die Mappen auf dem Tisch mit zusätzlichen Informationen „augmentiert“ werden sollen, ist eine genaue Positions- und Rotationsbestimmung notwendig. Optisches Tracking über eine Kamera ist dafür bestens geeignet. Hierfür wird das ARTag-Markersystem verwendet, da es relativ robust gegenüber Verdeckungen der Marker ist [13].

3.2.2 Direkte Ausgabe an den realen Mappen

Um die Suche nach realen Dokumenten und Mappen zu beschleunigen, soll an den Mappen ein Ausgabegerät angebracht werden, das diese nach Bedarf hervorheben kann. Da für das Hervorheben von Mappen eine binäre Ausgabe (ein oder aus) völlig genügt, ist für diese Aufgabe ein einfaches LED gut geeignet. Es ist am Karteireiter montiert und daher im Hängemappenständer gut erkennbar. Die Ansteuerung des LEDs geschieht ebenfalls über VoodooIO. Da bei VoodooIO jedes Gerät mit einer eindeutigen ID identifiziert wird, ist auch das LED identifizierbar und von außen ansteuerbar. Diese Lösung ist aber auf die Verwendung im Ständer eingeschränkt, da nur dort eine Verbindung mit VoodooIO und somit zu Strom besteht. Außerhalb des Ständers wäre eine kabellose Verbindung notwendig, welche VoodooIO aber nicht bietet.

3.2.3 Direkte Eingabe an den realen Mappen

Eine weitere Anforderung an das System ist, dass an den Mappen eine Eingabe zur Kommunikation mit dem DMS möglich ist. Genauso wie bei der Ausgabe muss dieses Gerät mit der Mappe eindeutig assoziierbar sein. Dieses Eingabegerät soll eine Navigation von der realen Mappe aus ermöglichen. Somit wurde ein kleiner Druckknopf, der am Karteireiter der Mappen neben dem LED angebracht ist, montiert. Dieser Knopf wird ebenfalls mit VoodooIO angesteuert und ist somit auch eindeutig zur Mappe zuordenbar. Auch wie das LED funktioniert der Druckknopf aber nur im Ständer und nicht am Tisch.

Für die Interaktion am Tisch wurde zusätzlich spezielles Papier aufge-

klebt. Basierend auf einer von Anoto[2] entwickelten Technologie kann dieses Papier als interaktive Benutzerschnittstelle verwendet werden. Aufgrund der Tatsache, dass die digitalen Stifte von Anoto eine drahtlose Kommunikation mit dem Computer erlauben, eignen sie sich hervorragend als Eingabegerät am Tisch.

3.2.4 Darstellung und Interaktion mit virtuellen Daten

Die Darstellung und Interaktion mit rein virtuellen Inhalten von Ordnern geschieht neben dem Computerarbeitsplatz auch direkt am Tisch. Die Anforderungen für diese Situation sind:

- Einfache Interaktion mit virtuellen Objekten
- Intuitiver Ladevorgang von virtuellen Objekten
- Robuste Tischoberfläche
- Stabiles Tracking
- Mehrbenutzerfähigkeit

Weiters soll die Tabletop Anwendung auch den allgemeinen *Design Guidelines für Tabletopdisplays* [33] entsprechen.

Um diesen Anforderungen gerecht zu werden, sind folgende Komponenten nötig:

- ein Display für die Darstellung von virtuellen Objekten
- ein System für die Erkennung von Mappen und somit das Laden von virtuellen Objekten
- ein System zur Interaktion mit virtuellen Objekten

Diese Komponenten haben starke Abhängigkeiten zueinander, vor allem, weil nicht alle verfügbaren Technologien miteinander kompatibel sind. So kann ein Trackingsystem, das die gesamte Tischfläche bedeckt, nicht mit einer Rückprojektion oder einem Plasma- oder LCD-Bildschirm verwendet werden.

Weiters ist auch zu beachten, dass nicht jedes Trackingsystem für den Tisch geeignet ist. Objekte mit denen keine Eingaben getätigt werden wollen (z.B. Tassen), dürfen die Interaktion nicht behindern. Zusätzlich müssen mehrere Personen gleichzeitig interagieren können und die gesamte Anwendung soll so natürlich wie möglich zu bedienen sein.

[2] http://www.anoto.com

Der *DiamondTouch* von MERL erfüllt diese Anforderungen sehr gut. Es werden nur Eingaben mit den Fingern erkannt und es können bis zu vier Personen gleichzeitig darauf arbeiten. Gegenstände werden vom Tracking ignoriert und beeinflussen die Anwendung somit nicht. Da die Eingabe über die direkte Berührung der Tischoberfläche funktioniert, ist auch eine möglichst natürliche Interaktion ohne zusätzlich benötigte Eingabegeräte möglich [23].

Die Darstellung virtueller Dokumente wurde über eine Top-Projektion realisiert, da die DiamondTouch-Oberfläche nicht transparent ist und somit eine andere Lösung nicht möglich ist.

Die Erkennung von Mappen wurde, wie schon vorher erwähnt, mittels ARTag realisiert. Diese Technologie ist mit dem DiamondTouch und der Top-Projektion kompatibel.

Neben der Hardware muss auch die Software den Tabletopanforderungen gerecht werden. Bei der Auswahl der richtigen Softwaretechnologien muss darauf geachtet werden, dass diese die simultane Interaktion von mehreren Benutzern und Eingabegeräten, und das Anpassen von virtuellen Darstellungen an die individuelle Sicht eines Benutzers (Rotation, Skalierung) unterstützen. Für diese Aufgabe wurden für solche Anforderungen entwickelte Bibliotheken verwendet. Die Bibliothek *JazzUp*[3] ermöglicht den simultanen Betrieb von mehreren Eingabegeräten und die darauf aufsetzende *Widget* Bibliothek[4] bietet ein grundlegendes, OpenGL-basiertes GUI-System, das beliebig rotierbare und skalierbare GUI-Elemente erlaubt.

3.2.5 Datenhaltung

Für die Datenhaltung am Computer soll in *Aroo* eine bestehende DMS Lösung verwendet werden. Ein weiteres Kriterium ist eine einfache Erweiterbarkeit und geringe Komplexität, damit eine rasche Integration möglich ist. Konkret muss es möglich sein, die Suchfunktion um die Ausgabe auf den realen Mappen zu erweitern und müssen die gespeicherten Daten von externen Anwendungen abrufbar sein und verändert werden können. Auch soll das Einrichten des DMS einfach und schnell möglich sein, damit der Zeitaufwand für die Installation des Systems verringert wird.

Aus diesen Anforderungen heraus wurde *Microsoft OneNote 2007*[5] als DMS Lösung gewählt. Es kann über *.NET* erweitert und auf gespeicherte Daten zugegriffen werden und es ist sehr einfach zu verwenden und einzurichten. Auch eine erweiterbare Suchfunktion ist über ein bereitgestelltes API verfügbar.

[3] http://sourceforge.net/projects/jazzup
[4] http://www.officeoftomorrow.org
[5] http://office.microsoft.com/onenote

3.3 Setup

Wie in Abbildung 3.2 zu sehen ist, besteht das Setup aus drei Hauptbestandteilen:

- ein Computerarbeitsplatz (a),
- ein erweiterter Hängemappenständer (b) und
- ein Besprechungstisch (c).

Der Computerarbeitsplatz besteht aus einem normalen Bürotisch, Sessel, Computer und Bildschirmen. Der Hängemappenständer ist zwischen Arbeitsplatz und Besprechungstisch positioniert. Er ist somit schnell von beiden Plätzen aus zu erreichen und die beinhalteten Mappen sind immer in Griffweite. Der dritte Teil, der Besprechungstisch, steht vor einer L-förmigen Couch, auf der drei Personen angenehm sitzen und gleichzeitig am Tisch arbeiten können. Eine vierte Person könnte an einer weiteren Seite auf einem Sessel sitzen. Der Tisch besteht aus einem Couchtisch an dessen Oberseite der *DiamondTouch* (DT) montiert ist und die gesamte Oberfläche bedeckt. Zusätzlich befinden sich über dem Tisch ein Projektor und eine Kamera für das Tracking. Der Projektor und die Kamera sind senkrecht auf einer Traverse montiert und auf den Tisch gerichtet (siehe Abschnitt 3.5.5).

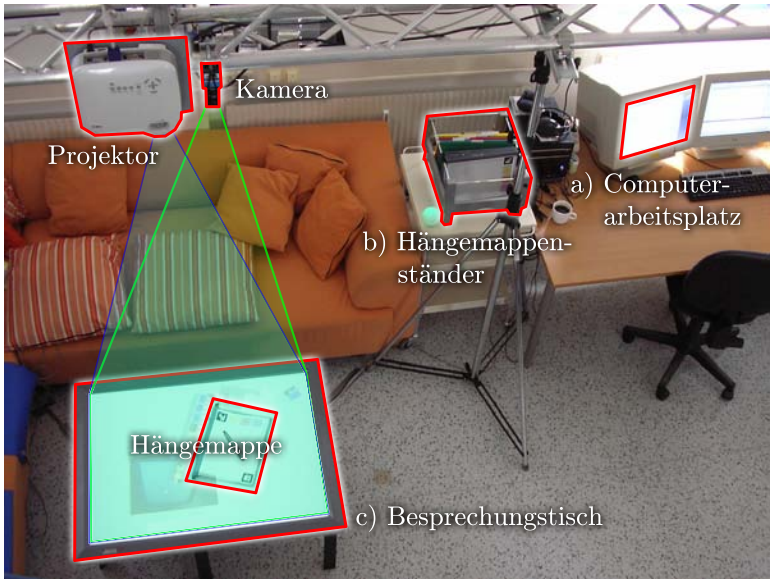

Abbildung 3.2: Das Setup besteht aus einem Computerarbeitsplatz (a), dem erweiterten Hängemappenständer (b), und dem digitalen Besprechungstisch (c).

3.4 Hardware Implementierung

3.4.1 Überblick

Die Hardwareimplementierung besteht aus den vier Kernkomponenten: Computer, Tisch, Hängemappenständer und Hängemappe (siehe Abb. 3.3). Im Zentrum steht der Computer, der durch die Verwendung verschiedener Technologien mit den anderen Komponenten verbunden ist. Über VoodooIO sind der Ständer und somit auch die in ihm beinhalteten Mappen verbunden. Zusätzlich sind die Mappen über Anoto-Technologie (digitale Stifte) drahtlos an den Computer angebunden. Zuletzt ermöglichen Projektor, Kamera und DiamondTouch die Verbindung vom Tisch zum Computer.

Die Erweiterungen der Hängemappenständer und der Hängemappen und der verbessertere VoodooIO-Controller basieren auf den Entwicklungen der „Smart Folders“[6].

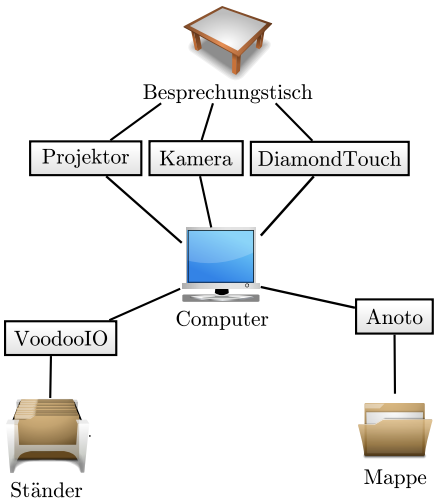

Abbildung 3.3: Das Zusammenspiel der Hardwarekomponenten: Die vier Kernkomponenten Computer, Tisch, Hängemappenständer und Hängemappe sind durch Verwendung unterschiedlicher Technologien (Kästen) miteinander verbunden.

3.4.2 Hängemappenständer

Der Hängemappenständer umfasst mehrere elektronische Geräte, wie einen kleinen Drückknopf und einen Drehknopf (siehe Abb. 3.4). Diese beiden Geräte sind mit dem VoodooIO-Controller über zwei Kabel verbunden. Der

[6] Die „Smart Folders“ sowie der VoodooIO Controller wurden von Matt Jervis (mattj@ihug.co.nz) entwickelt

Abbildung 3.4: Der Hängemappenständer: An der Vorderseite sind ein Druckknopf und ein Drehknopf angebracht.

Controller selbst ist über ein sechspoliges Patchkabel mit RJ25/6P6C Modulsteckern direkt mit dem Ständer verbunden. VoodooIO verwendet davon aber nur zwei Kontakte. Jeder einzelne Ständer wird somit von einem eigenen Controller gesteuert. Damit die Geräte am Ständer mit dem Computer kommunizieren können, ist der VoodooIO-Controller über ein USB Kabel am Computer angeschlossen.

Die vom Ständer beinhalteten Mappen sind ebenfalls mit einem VoodooIO-Geräte bestückt. Drückknopf und LED sind hierbei zu einem Gerät zusammengefasst und gelten somit als ein VoodooIO-Gerät. Um diese über den Computer ansprechen zu können, benötigen sie eine leitende Verbindung zum VoodooIO-Controller. Diese Verbindung wird über die Halteschienen des Ständers realisiert. Die Schienen sind aus Metall und somit elektrisch leitfähig. Dabei ist jeweils eine Schiene mit einem Kabel des VoodooIO-Netzwerkes verbunden. Bei den Mappen ist an den Unterseiten der Hacken je eine metallische Kontaktfläche angebracht. Diese sind über ein Kabel mit den LED und dem Drückknopf am Karteireiter verbunden. Wie in Abbildung 3.5 zu sehen ist, wird somit beim „Aufhängen“ über diese Schienen und den Hacken der Mappe eine elektrische Leitung hergestellt, über diese dann die Geräte kommunizieren können.

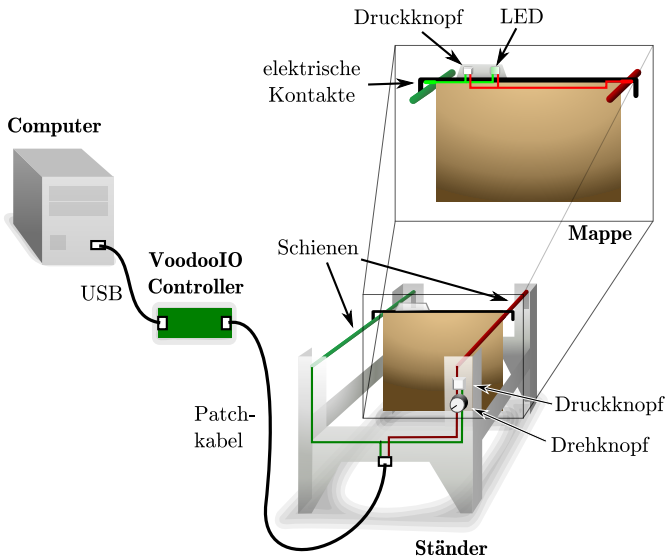

Abbildung 3.5: Über den Controller kann mit allen VoodooIO-Geräten kommuniziert werden. Die Mappen werden über elektrische Kontakte, die an der Unterseite der Hacken angebracht sind, beim Aufhängen im Ständer verbunden.

3.4.3 Hängemappe

Die Mappe selbst ist mit mehreren Technologien erweitert. Abbildung 3.7 zeigt wie VoodooIO, Anoto und ARTag verwendet werden. An der Oberseite der Mappe, am Karteireiter, sind über VoodooIO der Drückknopf und das LED angebracht (siehe Abb. 3.6). Diese beiden Geräte sind dabei als ein VoodooIO-Gerät, mit einer ID, zusammengefasst. Für VoodooIO ist somit ein Ordner ein eigenes Gerät mit einer Eingabe- und Ausgabefunktion.

Zusätzlich ist auf der Vorderseite der Mappe eine A4 Seite Anoto-Papier angebracht. Auf diesem „Digital Paper Interface“ [3] können die digitalen Stifte von Anoto beim Schreiben ihre eigene Position und die ID der Seite bestimmen. Für jede Mappe wird hierfür eine eindeutig identifizierbare Seite verwendet. Weiters sind diese Seiten auch für den Einsatz im sogenannten Streaming-Modus gedacht, d.h. Stifte speichern die gezeichneten Linien nicht wie üblich in den internen Speicher, sondern sie schicken ihre Position in Echtzeit über Bluetooth an den verbundenen Empfänger. Somit kann das Blatt Papier zusammen mit dem Stift als direktes Eingabegerät dienen.

Die Seite ist weiters noch in zwei verschiedene Bereiche gegliedert. Eine große freie Fläche in der Mitte ist für Notizen gedacht, die direkt mit der virtuellen Version der Mappe synchronisiert wird. Am rechten Rand sind drei Schaltflächen auf dem Papier aufgedruckt. Diese werden für die Steuerung der Darstellung von virtuellen Inhalten am Tisch verwendet (Kapitel 4).

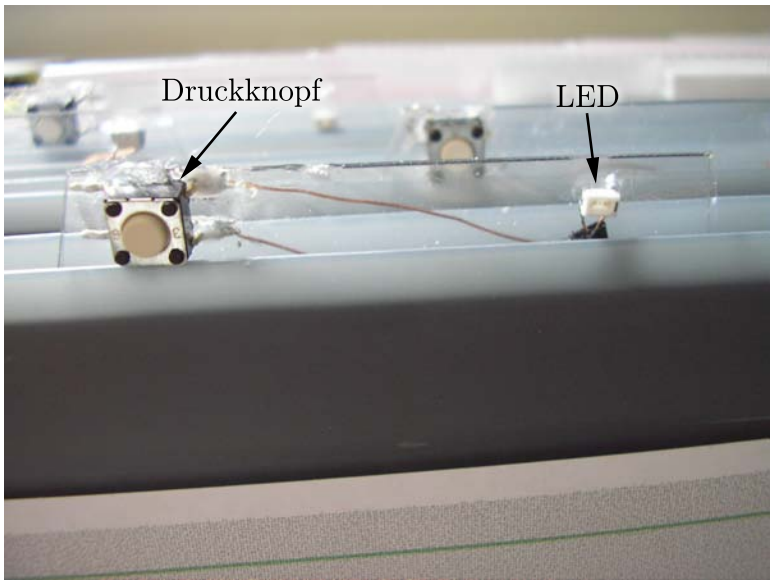

Abbildung 3.6: Am Karteireiter der Mappe sind ein Druckknopf und ein LED angebracht.

Zwei ARTag-Marker werden zur Positions- und Rotationsbestimmung der Mappen am Tisch verwendet. Auf jeder einzelnen Mappe sind jeweils unterschiedliche Marker gedruckt, um die Identifizierung am Tisch zu gewährleisten. Sie sind an den beiden gegenüberliegenden Ecken der Seite angebracht, damit das Tracking soweit wie möglich stabilisiert werden kann (siehe Abschnitt 5.7.3). Die Marker selbst haben eine Seitenlänge von 40 mm, um noch gut von der Kamera erfasst werden zu können. Dieser Wert hat sich durch Tests verschiedener Markergrößen (30, 40 und 50 mm) ergeben und war der kleinste, bei dem die Erkennung ohne Fehler[7] funktioniert hat.

3.4.4 Tisch

Der zum Einsatz kommende DiamondTouch ist der *DT2-107* von MERL[8] und ist mit vier Sitzmatten ausgestattet. Somit können bis zu vier Personen gleichzeitig damit arbeiten. Die Sitzmatten sind über ein Kabel mit der Elektronik an der Unterseite des DT verbunden [7]. Aus optischen Gesichtspunkten wurden diese Matten in die Polster der Sitzgarnitur eingearbeitet. Das zusätzliche Material zwischen der Matte und der darauf sitzenden Person wirkt sich zwar auf die Intensität des vom Tisch gemessenen elektrischen Feldes aus, die Qualität des Trackings wird dabei aber nur minimal beeinträchtigt.

[7] In einer kurzen Testsequenz wurden alle Marker in jedem Frame richtig erkannt.
[8] http://www.merl.com/projects/DiamondTouch/

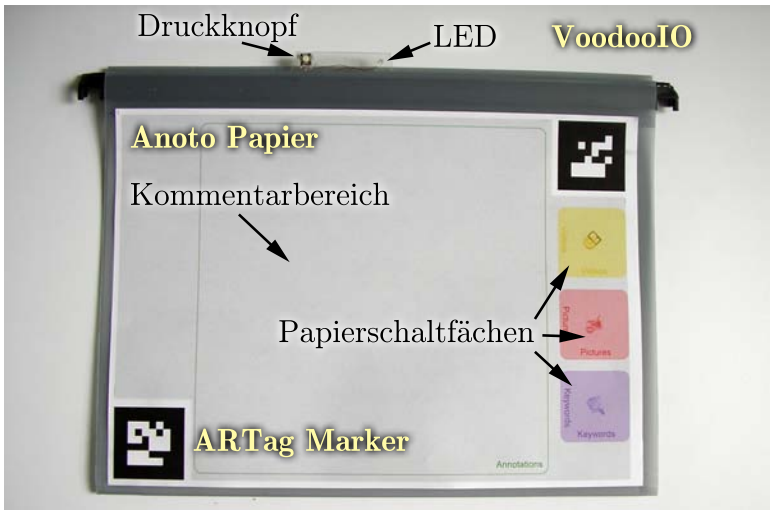

Abbildung 3.7: Die Hängemappe wird mit VoodooIO-Geräten, Anoto-Papier und ARTag-Markern erweitert. Der Druckknopf und das LED werden über VoodooIO angesprochen, der Kommentarbereich und die Papierschaltflächen mit Anoto und die ARTag wird zum Tracken auf dem Tisch verwendet.

Der DT ist auf einem niedrigen Couchtisch montiert. Von oben projiziert der NEC VT670 Projektor mit einer Auflösung von 1024x768 auf die berührungssensitive Oberfläche.

Neben dem Projektor ist die ImagingSource DFK 31AF03 Kamera mit der Auflösung von ebenfalls 1024x768 montiert und per Firewire[9] an den Computer angebunden. Die Kamera erfasst dabei einen größeren Ausschnitt, die nahe Umgebung des Tisches bis in eine Entfernung von 10 bis 20 cm vom Tischrand, als die Projektion. Somit wird eine Mappe schon in der Nähe des Tisches erkannt und nicht nur auf der Projektionsfläche.

3.5 Software Implementation

3.5.1 Überblick

Die Softwareimplementierung besteht aus zwei Hauptkomponenten:

- Eine OpenGL-basierte Anwendung (**Tischanwendung**), die für die Tischdarstellung zuständig ist, und
- ein **Add-in für OneNote**.

Die *Tischanwendung* stellt rein virtuelle Inhalte mittels OpenGL dar und ermöglicht die Mehrbenutzerinteraktion über den DiamondTouch. Neben

[9] IEEE 1394

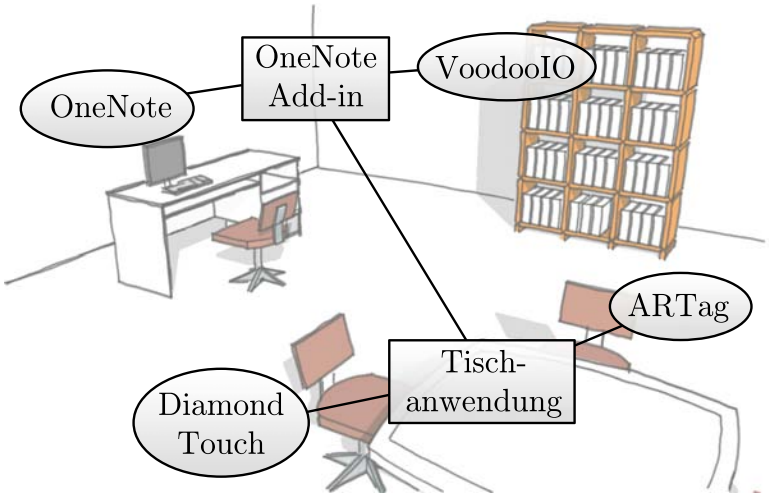

Abbildung 3.8: Die Softwareimplementierung ist auf eine Einzelbenutzeranwendung am Arbeitsplatz (OneNote Add-in) und eine Mehrbenutzeranwendung am Besprechungstisch (Tischanwendung) aufgeteilt.

dieser Aufgabe werden in dieser Anwendung über die Kamera auch Mappen am Tisch erkannt und dazugehörende Bilder und Videos aus OneNote geladen. Es werden die für diesen Zweck in C++ entwickelten Bibliotheken, *JazzUp* und *Widget* verwendet.

Das *Add-in für OneNote* ist nur für die Benutzung mit einem Benutzer gedacht und somit eine normale Desktopanwendung. Die Einbindung in OneNote geschieht über ein *Toolbar-Add-in*, mit der eigene Funktionalitäten über eine zusätzliche Schaltfläche in der Toolbar integrierbar sind. Dieses in C# entwickelte Addin bietet eine erweiterte Suchfunktion und integriert die Navigation von OneNote und den realen Ordnersystem.

Technologien

Die Implementierung von *Aroo* verwendet folgende Softwarekomponenten und Technologien:

API/Bibliothek	**Version**	**Quelle**
Microsoft OneNote 2007 API	v1.1.4322	MSDN[10]
VoodooIO SDK	SVN Rev-254	www.voodooio.com
ARTag	Rev 2	www.artag.net
DiamondTouch SDK	2.1	MERL Projects[11]
AnotoPen	0.5.0.0	OOT[12]

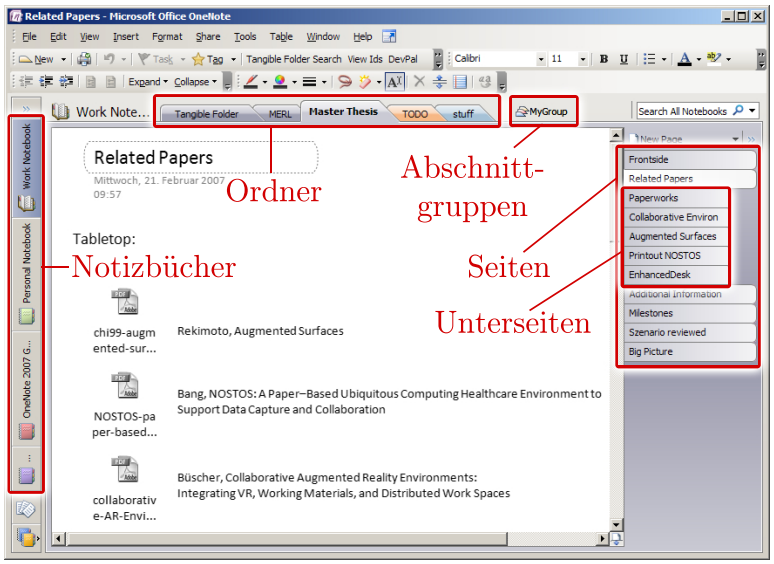

Abbildung 3.9: In OneNote sind Dokumente in Notizbüchern, Ordner, Abschnittgruppen, Seiten und Unterseiten geordnet.

Die realen Mappen im Hängemappenständer sind mit den Daten aus Microsoft OneNote 2007 verknüpft. OneNote eignet sich sehr gut zum Schreiben von Notizen, Sammeln und Suchen von Informationen und netzwerkunterstützter Teamarbeit. Aus diesen Gründen wird es für das Management, Suchen und Ablegen von Dokumenten in unserem Setup verwendet.

Über das VoodooIO SDK können alle physikalischen Eingabegeräte auf den Mappen und dem Ständer leicht angesprochen und auf Eingaben von diesen Geräten reagiert werden. Zusätzlich zu VoodooIO wird die Kommunikation durch die Verwendung von Anoto-Technolgie ergänzt. Zuletzt wird über ARTag-Tracking und DiamondTouch die intuitive Interaktion am Tisch ermöglicht.

3.5.2 Technologien

OneNote

OneNote verwaltet für *Aroo* alle Dokumente und Daten. Es ist hier aber nicht nur für die Datenhaltung und die Organisation von Dokumenten zuständig, sondern stellt auch einige weitere wichtige Funktionalitäten zur Verfügung.

Das Hauptmerkmal an OneNote ist die Art und Weise, wie Dokumente organisiert werden. Anders als in den meisten DMS, sind Dokumente nicht

[10] http://msdn2.microsoft.com/en-us/office/ms406042.aspx
[11] http://www.merl.com/projects/dtsdk/
[12] http://www.officeoftomorrow.org

als Dateien in Ordnern, sondern auf Seiten gespeichert. Die Organisation ist somit auch nicht strikt hierarchisch, sondern weniger stark reglementiert als bei vielen anderen DMS. Dokumente sind auf einer Seite nicht nach einer Reihenfolge geordnet, sondern nach einem „räumlichen“ Schema.

In OneNote können verschiedenste Inhalte abgespeichert werden. Wie in jedem DMS können Dateien abgespeichert werden. Das Dateiformat ist in diesem Fall unbedeutend und somit kann jedes gängige Dokumenten- und auch Bildformat abgelegt werden.

In vielen DMS werden diese Dateien aber nicht weiter verarbeitet. In OneNote dagegen gibt es die Möglichkeit, nicht nur ein Symbol und den Dateinamen anzuzeigen, sondern auch deren Inhalt. Dies gilt für alle Dokumente, für die im Betriebsystem eine Druckfunktion zur Verfügung steht. Ein Beispiel für solche Dateien ist PDF. Wenn Adobe Reader installiert ist, können diese Dokumente als sogenannter Ausdruck eingefügt werden. Jede ausgedruckte Seite wird dabei als Bild eingefügt. Somit kann man den Inhalt von Dokumenten auch gleich in OneNote betrachten, ohne eine externe Anwendung starten zu müssen. Weiters ermöglicht dieser virtuelle Ausdruck auch das Kommentieren von Dokumenten direkt in OneNote.

Erstellte Kommentare sind ein eigener Inhaltstyp von OneNote. Alle Dokumente und jede leere Fläche können mit der Maus oder mit einem digitalen Stift (z.B. am TabletPC) annotiert werden. Diese Notizen werden als Vektorgrafiken gespeichert und können nachträglich verändert werden. Sie werden nicht mit dem darunterliegenden Objekt verbunden, sondern liegen als eigene Ebene über diesen.

Zusätzlich zu diesen beiden Typen kann auch formatierter Text an beliebiger Position eingefügt werden. Dabei kann Text entweder direkt über die Tastatur eingetippt werden oder über Copy&Paste aus einer externen Anwendung kopiert werden. Hierbei wird nicht nur der Text kopiert, sondern auch die Herkunft des Textes. Wird zum Beispiel Text aus einer Webseite in OneNote kopiert, so wird auch die URL der Internetseite mitgespeichert.

Weiters können auch Audio- und Videoinhalte eingefügt oder direkt aus OneNote aufgenommen werden. Diese werden wie normale Dateien nur mit Symbol und Titel angezeigt und können bei Bedarf auch abgespielt werden.

Wie schon erwähnt, verwendet OneNote Seiten zum Ablegen von Dokumenten und andern Inhalten. Wie in Abbildung 3.9 dargestellt, sind Dokumente über die Hierarchiestufen Notizbücher, Abschnittgruppen, Ordner, Seiten und Unterseiten geordnet. Diese Hierarchie wird aber zum Teil aufgebrochen. So können *Notizbücher* sowohl *Abschnittgruppen* als auch *Ordner* beinhalten. Ein hierarchischer Zusammenhang zwischen *Seite* und *Unterseite* ist überhaupt nicht vorhanden. Beide werden nebeneinander in *Ordern* gespeichert und unterscheiden sich nur in einer unterschiedlichen Ansicht in der Seitenliste des *Ordners*.

Eine Besonderheit an OneNote ist die Suchfunktion. Es kann nicht nur Text durchsucht werden, sondern auch Bilder, Handschrift und Audiospuren.

Bilder und auch die vorher erwähnten Ausdrucke werden beim Einfügen mit einer *Optical Character Recognizion* (OCR) Software gescannt. Die beinhalteten Texte werden extrahiert und mit ihrer Position im Dokument gespeichert. So kann beim Anzeigen des Suchergebnisses, der gesuchte Begriff auch auf Bildern hervorgehoben werden. Zusätzlich kann man auch handschriftliche Notizen in Text umwandeln oder im Hintergrund indizieren lassen. Darüber hinaus kann OneNote auch die gesprochene Sprache in Audiodateien erkennen und somit das Suchergebnis um diese erweitern.

VoodooIO SDK

Das VoodooIO SDK ermöglicht die Kommunikation mit den über USB angeschlossenen VoodooIO-Geräten. Das SDK verwendet dabei eine Server-Client Architektur, wobei die Kommunikation über TCP/IP geschieht. Der Server ist als eigenständige Anwendung verfügbar. Die Einbindung in eine eigene Anwendung geschieht über den Client, der als eine .NET Bibliothek vorliegt. Beide Teile sind aber noch in Entwicklung und laufen noch nicht stabil.

ARTag

ARTag ist ein markerbasiertes optisches Trackingsystem, das, ähnlich wie ARToolkit [18], quadratische, schwarz-weiße Marker in einem Kamerabild erkennt und deren Position im dreidimensionalen Raum bestimmt. Im Unterschied zu ARToolkit ist der Innenbereich des Markers nicht frei definierbar, sondern eine 6x6 große Codematrix, in der die eindeutige Marker-ID gespeichert ist. Diese ID ist mit Fehlerkorrekturmaßnahmen kodiert und kann somit auch bei teilweiser Überdeckung noch berechnet werden. Ein weiterer Vorteil im Vergleich zu ARToolkit ist die höhere Robustheit gegenüber Beleuchtungsunterschieden im Kamerabild und die stabilere Erkennung der Marker bei verdeckten Rändern [13].

DiamondTouch

Der MERL DiamondTouch (DT) Tisch ist eine mehrbenutzerfähige, berührungssensitive Trackingoberfläche. Die Besonderheit an dieser Technologie ist, dass zum einen *systemfremde* Objekte (z.B. Tassen, Dokumente) nicht erkannt werden und zum anderen jeder Berührungspunkt am Tisch einem Benutzer eindeutig zugeordnet werden kann. Wie in Abbildung 3.10 dargestellt wird, muss zur Erkennung von Berührungen jeder Benutzer auf einer Matte sitzen. Diese sind über ein Kabel mit dem DT verbunden. Berührt ein Finger die Oberfläche wird über ein Netz von Antennen der Berührungspunkt erkannt. Der DT ist über USB am Computer angeschlossen und benötigt sonst keine weitere Stromverbindung [7].

Zur Entwicklung eigener Anwendungen stellt MERL, neben einer einfachen Mausemulation für Einzelbenutzeranwendungen, ein umfangreiches SDK zur Verfügung. Neben Bibliotheken für C/C++ stehen auch Portierungen für Java und .net zur Verfügung. Die Kalibirierung des Tisches geschieht schon auf Treiberebene und muss daher nicht extra berücksichtigt werden.

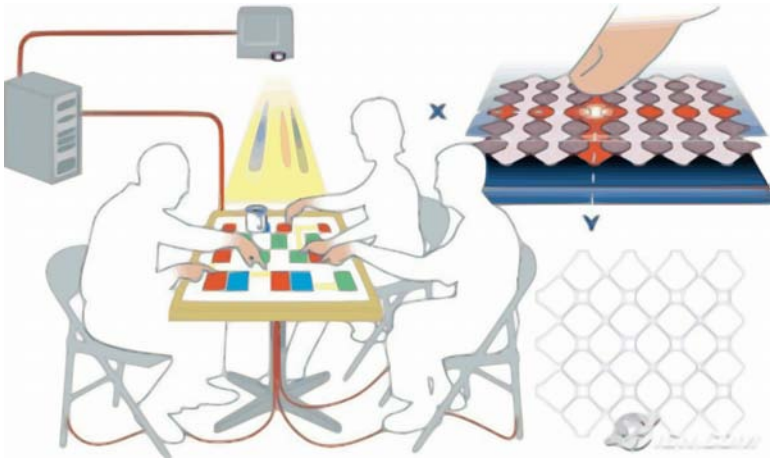

Abbildung 3.10: Mit einem Netz aus diamantförmigen Antennen erkennt DiamondTouch Berührungen an der Oberfläche. [7].

Anoto

Anoto-Stifte verwenden zur Berechnung ihrer eigenen Position eine eingebaute Kamera und ein auf das Papier gedrucktes Punktmuster (siehe Abb. 3.11). Das Papier ist hierbei der Informationsträger, auf dessen hochauflösend gedrucktem Punktmuster die Positionsinformationen kodiert sind. Der Stift liest über die Kamera diese Information aus und braucht dabei 6 x 6 Punkte, die eine Fläche von 1,5 x 1,5 mm einnehmen, um die Position des Stiftes auf einen Millimeter genau zu bestimmen [27]. Die in *Aroo* verwendeten Stifte von Anoto verwenden dabei eine Kamera, die das Papier mit einer Frequenz von 75Hz abtastet [26].

Die digitalen Stifte können in zwei verschiedenen Betriebsmodi verwendet werden. Einerseits können sie die erkannten Striche in einem internen Speicher ablegen und zu einem vom Benutzer gewünschten Zeitpunkt in einem Stück übertragen. Andererseits können sie jede Information in Echtzeit über eine Bluetoothverbindung schicken. Welcher dieser Modi verwendet wird, hängt nur von dem gedruckten Punktmuster der Seite ab. Nur bestimmte Seiten aktivieren die Echtzeitübertragung, alle anderen arbeiten nur im offline Modus. In *Aroo* werden die Stifte im Echtzeitübertragungsmodus verwendet.

Die Integrierung der über Bluetooth in Echtzeit übertragenen Informationen in eine eigene Anwendung ermöglicht die auf *JazzUp* basierende *AnotoPen* Bibliothek. Diese C++ Bibliothek ermöglicht die gleichzeitige Verwendung mehrerer digitaler Stifte und das Übertragen aller Events über Netzwerk mit UDP Packeten.[13]

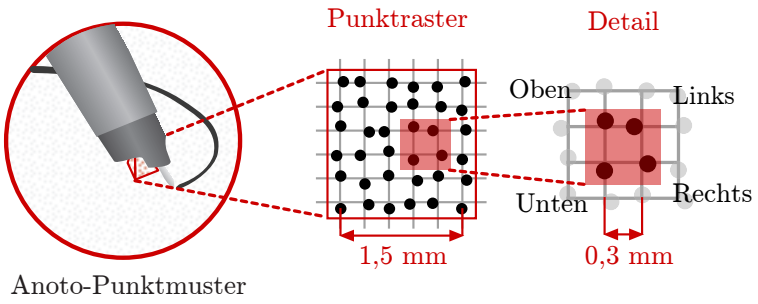

Abbildung 3.11: Die Funktionsweise der Anototechnologie.

3.5.3 Mapping

Damit eine Mappe an den unterschiedlichen Orten (Hängemappenständer, Tisch, Arbeitsplatz) identifiziert werden kann, ist dieser für jeden Ort und Verwendungszweck mit einer ID ausgestattet. Jede Mappe ist durch folgende IDs identifiziert:

- OneNote-Section-ID,
- Anoto-Page-ID,
- VoodooIO-Geräte-ID und
- ARTag-Marker-ID

Jede reale Mappe wird mit einer Section von OneNote assoziiert, da diese am meisten einer realen Mappe ähnelt. Ähnlich wie eine reale Mappe beinhaltet diese viele Seiten, und es gibt keine weitere Hierarchiestufe darunter.

Anders als die *OneNote-Section-ID* und die *Anoto-Page-ID* sind die beiden anderen IDs an einen bestimmten Ort gebunden. Die *ARTag-Marker-ID* kann nur am Tisch verwendet werden, und die *VoodooIO-Geräte-ID* nur im Ständer. Somit ist die reale Mappe gleich durch drei IDs gekennzeichnet

[13] Anmerkung: Zum Zeitpunkt der Erstellung dieser Arbeit gab es noch keine Softwareunterstützung für die Echtzeitfunktion der digitalen Stifte zur Entwicklung von Anwendungen von Seiten von Anoto. Außer einem SDK von Maxell, die aber nur einen Stift unterstützte, gab es nur Eigentwicklungen.

(Anoto, ARTag, VoodooIO), während der virtuelle Ordner nur durch die Section ID repräsentiert wird.

In dieser Implementierung ist dieses Mapping in einer separaten XML Datei abgespeichert, die in beiden Anwendungen geladen wird. Im Wurzelelement `<TangibleFolderMapping>` werden alle zusammengehörigen Identifikationsbezeichnungen in dem Element `<folder>` als Attribute eingetragen. Diese Attribute sind:

- `sectionId`: objectID der OneNote Section
- `voodooIOId`: VoodooIO-GeräteID der zugehörigen Mappe
- `frontMarkerId`: ARTag-Marker/Object-Bezeichnung der Marker an der Vorderseite der Mappe
- `paperAreaId`: Page-ID der Fläche für Notizen auf dem Anotopapier.

3.5.4 OneNote Add-in

Obwohl die Möglichkeit der Entwicklung eines Add-ins für OneNote, über die Einbindung einer eigenen Schaltfläche, auf der MSDN Webseite publiziert worden ist [4], gibt es keine offizielle Dokumentation darüber. Die folgende Beschreibung des Interfaces für OneNote Addins stammt daher aus dem Weblog „The Unknown OneNote Guy's Blog“[14] und dem privaten Weblog von Daniel Escapa[15].

OneNote Add-in Interface

Add-ins für OneNote 2007 haben sehr beschränkte Möglichkeiten. Es können keine Menüs oder Toolbars erstellt werden, sondern nur das Einbinden einer einzigen Schaltfläche pro Add-in in eine Toolbar ist möglich.

Für die Entwicklung eines Add-ins ist ein sehr einfaches Interface mit nur zwei Methoden bereitgestellt. Um dieses Interface in C# verwenden und eine eigene Klasse ableiten zu können (Code siehe Anhang A.1), muss es zuvor aus *Component Object Model (COM)* importiert werden. Dazu muss das Interface `IOneNoteAddIn` im Quellcode wie folgt definiert werden:

```
1 [ComImport, Guid("C9590FA7-2132-47fb-9A78-AF0BF19AF4E6")]
2 public interface IOneNoteAddIn
3 {
4   bool OnClick([In] string strActivePageID);
5   bool OnEvent([In] OneNoteAddIn_Event evt,
6                [In] string strParameter);
7 }
```

Code 3.1: OneNoteAddin Interface

[14] http://unknown-onenote.blogspot.com
[15] http://blogs.msdn.com/descapa/

Das Interface `IOneNoteAddIn` besitzt zwei abstrakte Methoden `OnClick` und `OnEvent`:

- `OnClick` wird beim Klicken auf die Schaltfläche des Add-ins von OneNote aufgerufen. Als einziger Parameter wird die ID der aktuell angezeigten Seite übergeben.
- Über die Methode `OnEvent` kann ein Add-in registrierte Events empfangen. Es werden dabei nur jene Events übertragen, die in der Registry eingetragen sind. Der erste Parameter `evt` der Methode identifiziert dabei das Event und der zweite `strParameter` übergibt zusätzliche Eventdaten als Zeichenkette.

Über den Rückgabewert beider Funktionen kann die Schaltfläche deaktiviert (Rückgabewert = false) oder aktiviert werden.

Implementierung des Add-ins

Eigene Add-ins müssen von dem Interface `IOneNoteAddIn` abgeleitet werden. Dabei ist der vor der Klassendefinition (siehe Code 3.2 Zeile 4) definierte *Global Unique Identifier (GUID)* [22] wichtig. Diese Zeile definiert eine eindeutige GUID für ein COM-Objekt. Das ist notwendig, weil OneNote Addins über COM lädt und somit auch jedes Addin ein COM-Objekt sein muss. Diese GUID wird später bei der Registrierung des Addins wieder benötigt. Für jedes Addin ist dabei eine eigene GUID zu generieren[16].

Weiters sind im Code 3.2 auch die beiden Eventhandler mit einer minimalen Funktionalität implementiert worden. Zu beachten ist, dass beide immer den Wert `true` rückliefern, damit die Schaltfläche nicht deaktiviert wird.

OneNote XML

Für das Auslesen, Einfügen und Verändern von Daten wird vom OneNote API ein eigener XML Dialekt [4] verwendet. Diese XML-Daten werden von OneNote aber nur zur Kommunikation verwendet, nicht in der Datenspeicherung selbst.

Das OneNote 2007 XML-Schema beschreibt die gesamte Hierarchie (Notebook, Sectiongroup, Section und Page) wie auch den Seiteninhalt selbst. Die Tags in Abbildung 3.12 beschreiben die Hierarchie von OneNote und können im XML-Text auch als Wurzelelemente verwendet werden. Die Identifizierung aller Hierarchieelemente geschieht über das Attribut `ID`. Diese ID wird auch benötigt, wenn man auf ein bestimmtes Element über das OneNote API zugreifen will. In *Aroo* wird die `ID` des Elements `Section` auch

[16] Eine neue GUID kann über das Visual Studio Tool „guidgen.exe“ erzeugt werden, wobei „Registryformat“ ausgewählt werden muss

```
1  namespace OneNoteSampleAddin
2  {
3    // Definition der GUID dieses Addins
4    [Guid("87E61590-99E0-46d3-99F1-15B95092CF13")]
5    public class AddInSample : IOneNoteAddIn
6    {
7      public bool OnClick([In] string strActivePageID)
8      {
9        // Ausgabe der aktuellen Seiten ID
10       MessageBox.Show("PageId:␣" + strActivePageID);
11       return true;
12     }
13
14     public bool OnEvent([In] OneNote.OneNoteAddIn_Event evt,
15                         [In] string strParameter)
16     {
17       // nichts zu tun
18       return true;
19     }
20   }
21 }
```

Code 3.2: OneNoteAddin Interface Implementierung

zum Mapping der virtuellen Ordnern mit den realen Mappen verwendet (siehe Abschnitt 3.5.3).

Ein Beispiel für die Verwendung der Hierarchieinformation ist das Abfragen der von OneNote aktuell angezeigten Section. Mit Hilfe der Methode `GetHierarchy()` wird die gesamte Hierarchie (alle Notebooks und Sections) bis zu den Seiteninformationen ausgelesen (vergleiche Code 3.3). Die in `strXML` ausgelesenen Daten werden in diesem Fall in den XML-Parser von .NET eingelesen. Danach wird die aktuell angezeigte Section mittels einer XPath-Query [5] gesucht (vergleiche Code 3.3 Zeile 22).

Wie auch die Hierarchie wird der Seiteninhalt durch XML-Elemente beschrieben. In Abbildung 3.13 sind diese Elemente aufgelistet. Je nach Inhaltstyp werden Daten binär oder textuell bzw. in die XML-Datei eingebettet oder als Verknüpfung eingefügt. Wie im folgenden Beispiel zu sehen ist, werden Dateien (`InsertedFile` oder `MediaFile`) üblicherweise als Verknüpfung eingefügt. Sowohl `pathCache` als auch `pathSource` verweisen auf eine Datei im Dateisystem.

```
1 <one:InsertedFile
2      pathCache="C:\SampleDir\SampleFile.pdf"
3      pathSource="C:\SampleDir\SampleFile.pdf"
4      preferredName="SampleFile" />
```

Code 3.4: XML-Code einer PDF-Datei

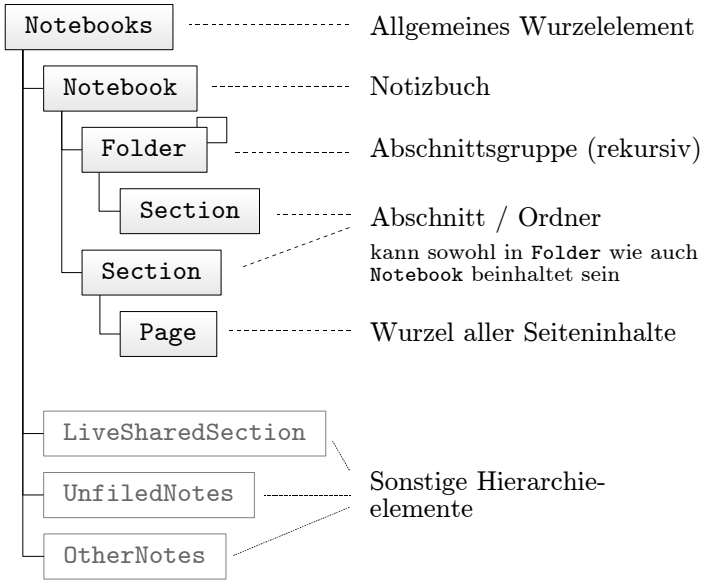

Abbildung 3.12: Die Darstellung der Hierarchie von OneNote im XML Schema.

Sind die Daten über Verknüpfungen eingebunden, so kann dieses Format sowohl beim Einfügen von neuen Elementen wie auch beim Auslesen verwendet werden. Anders ist das bei eingebundenen binären Daten, wie zum Beispiel bei Skizzen (`InkDrawing`). Beim Auslesen kann auf die binären Daten über die sogenannte `callbackID` zugegriffen werden. Das OneNote API stellt für solche Zugriffe eine eigene Methode zum Lesen von binären Daten zur Verfügung.

```
1 <one:InkDrawing lastModifiedTime="..." objectID="...">
2   <one:CallbackID
3    callbackID="{05BE85A3-39E6-4A3B-BDD0-DDC3B6BDAF53}{42}{B0}"/>
4 </one:InkDrawing>
```

Code 3.5: XML-Code einer Skizze beim Auslesen aus OneNote

```
1 <one:InkDrawing>
2   <one:Data>
3         <![CDATA[ ... ]]>
4   </one:Data>
5 </one:InkDrawing>
```

Code 3.6: XML-Code einer Skizze beim Einfügen in OneNote

```
1 // Initialisieren der OneNote API
2 OneNote.Application onApp = new OneNote.Application();
3
4 ...
5
6 // Auslesen der OneNote Hierarchie von der Wurzel bis zu den
7 // Seiten
8 string strXML;
9 onApp.GetHierarchy(null, // null = Wurzel
10                   OneNote.HierarchyScope.hsPages, // Seiten
11                   out strXML); // genierierter XML-Text
12
13 // Verarbeitung der XML-Daten
14 XmlDocument xmlDoc = new XmlDocument();
15 // Laden des XML-Textes
16 xmlDoc.LoadXml(strXML);
17
18 ...
19
20 // Absetzen einer XPath-Query (Selektiere das erste Element
21 // dessen Attribut "`isCurrentlyViewed"' den Wert "`true"' hat.
22 string xpQuery = "//one:Section[@isCurrentlyViewed=\"true\"]";
23 XmlElement xmlActiveSection =
24    (XmlElement)xmlDoc.SelectSingleNode(xpQuery);
25 // Auslesen des Attributs ID
26 string strActivePageID = xmlActiveSection.GetAttribute("ID");
```

Code 3.3: Auslesen der ID der aktuell angezeigten Section.

Beim Einfügen kann `callbackID` nicht verwendet werden. In diesem Fall müssen die binären Daten in XML-Code base64-kodiert eingefügt werden. Wie im folgenden Beispiel müssen diese Daten mit den Tag `Data` in einer Zeichendatensektion (`CDATA`) abgelegt werden.

Generell müssen beim Einfügen von Daten nicht immer alle Attribute gesetzt werden. Beispielsweise werden alle IDs und auch Größe und Position von Elementen automatisch generiert oder berechnet.

Registrierung des Add-ins

Damit OneNote ein Add-in einbinden kann, muss dieses in OneNote registriert werden. Hier kommt die GUID des Add-ins zum Tragen, denn OneNote benötigt diese, um es zu laden. Diese Information erhält OneNote über die *Windows Registry*, in dem das Add-in eingetragen werden muss. Zusätzlich werden über die Registry auch einige Eigenschaften gesetzt. So werden hier die Events aktiviert und das Erscheinungsbild der Schaltfläche bestimmt.

Unter dem Eintrag `HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\OneNote\AddIns` sind alle OneNote Add-ins eingetragen. Jedes besitzt

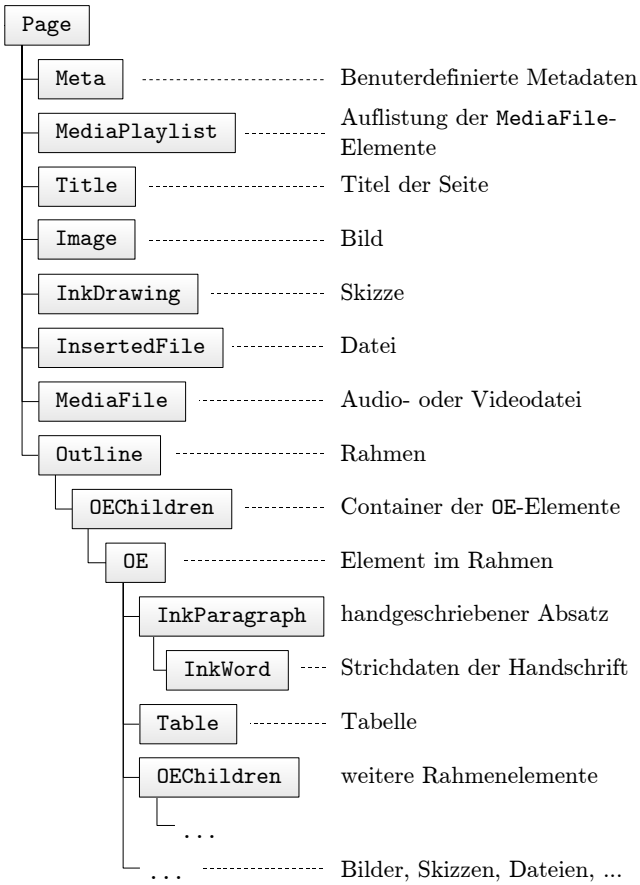

Abbildung 3.13: Die Kinder des `Page`-Elements stellen den Seiteninhalt (Skizzen, Dateien, Videos, usw.) dar.

einen eigenen Ordner, der als Name die GUID des Add-ins tragen muss. In Tabelle 3.1 sind alle notwendigen und möglichen Einträge aufgelistet. Abbildung 3.14 zeigt den Registry-Eintrag des Beispiel-Add-ins.

VoodooIO Einbindung

Die Software zur Ansteuerung von VoodooIO besteht aus dem *Service*, der als ein eigenständiges Programm ausgeführt wird und aus dem *Proxy*, der in die Anwendung integriert wird. Der Proxy ist in eine .NET Bibliothek und kann so in eigene Anwendungen integriert werden.

Die Kommunikation zwischen beiden geschieht über TCP/IP. Der Service agiert als Server und kann somit mehrere Proxies als Client bedienen. Das ermöglicht die parallele Verwendung mehrerer VoodooIO-Anwendungen. (siehe Abb. 3.15).

Name	Typ	Beschreibung
`Name`*	String	Name des Add-ins
`ButtonText`	String	Text auf der Schaltfläche
`CLSID`*	String	GUID des Add-ins
`CommandBar`	DWord	Ort wo die Schaltfläche erzeugt wird
`IconPath`	String	Pfad zu einem Icon (bmp, dll oder exe)
`IconIndex`	DWord	ResourceID des Icons im `IconPath`
`LoadOnNewWindow`	DWord	$\neq 0$: Add-in beim Onenote-Start laden
`UnloadAfterClick`	DWord	$\neq 0$: Add-in nach einem Klick freigeben
`EventDesired_HierarchyChange`	DWord	$\neq 0$: HierarchyChange-Events empfangen
`EventDesired_Navigation`	DWord	$\neq 0$: Navigation-Events empfangen

Tabelle 3.1: Registry-Einträge für ein OneNote Add-in. Alle mit * gekennzeichneten Einträge sind obligat. In `CommandBar` können die Werte von 0 bis 9 eingetragen werden [11]. Jede steht dabei für eine andere Toolbar.

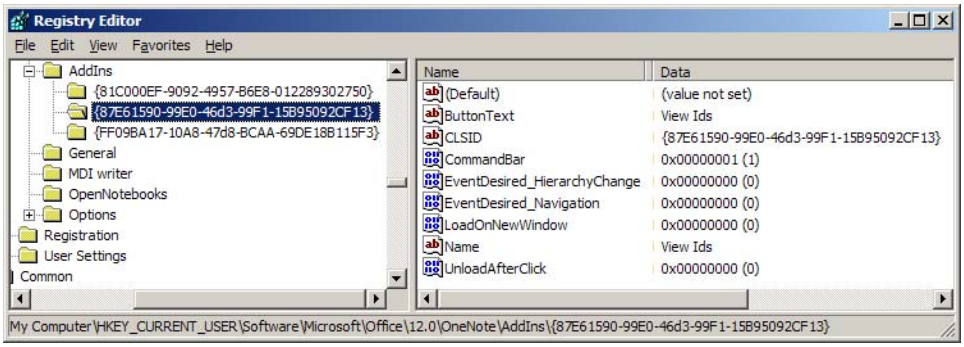

Abbildung 3.14: Der Registry-Eintrag für das Beispiel-Add-in.

Für die Einbindung des Proxies in eine Anwendung muss ein Objekt der Klasse `VIOServiceProxy` instanziert und dieser zum Service verbunden werden (Code 3.7 Zeile 2 und 4). Jeder Proxy benötigt eine eindeutig identifizierbare Kennung, die dem Konstruktor als Zeichenzette übergeben werden muss (`"VoodooIOSample"`). Neben diesen Schritten ist im Codeabschnitt 3.7 auch die notwendige Verbindung der Events zu den Eventhandlermethoden dargestellt. Die Bezeichnung „Node“ steht in diesem Kontext für ein VoodooIO Gerät.

Der Proxy stellt insgesamt vier verschiedene Events zur Verfügung:

- `NodeAdded`: Ein Gerät wurde hinzugefügt.
- `NodeRemoved`: Ein Gerät wurde entfernt.
- `NodeManipulated`: Eine Eingabe wurde an einem Gerät getätigt.
- `Disconnected`: Die Verbindung zum Service wurde getrennt.

```
1 // Erstellen eines neuen Proxy-Objekts
2 VIOServiceProxy proxy = new VIOServiceProxy("VoodooIOSample");
3 // Herstellen einer Verbindung zum Service
4 // (5555 = VoodooIO Standardport)
5 proxy.Connect("localhost", 5555);
6 // Verbindung zu den Events
7 proxy.NodeManipulated +=
8   new VIOServiceProxy.NodeManipulatedHandler(OnManipulated);
9 proxy.NodeAdded +=
10   new VIOServiceProxy.NodeAddedHandler(OnAdded);
11 proxy.NodeRemoved +=
12   new VIOServiceProxy.NodeRemovedHandler(OnRemoved);
```

Code 3.7: Verwendung des VoodooIO Proxies

Die Kommunikation von der Anwendung zu den VoodooIO-Geräten erfolgt über `VIONode` Objekte. Auf diese kann entweder über die Eventargumente von `NodeAdded`, `NodeRemoved` und `NodeManipulated` zugegriffen werden oder, wenn die ID des Gerätes bekannt ist, über die Methode `VIONode.getNode()`.

Jedes VoodooIO-Gerät kann über mehrere Eingaben (Inputs) und Ausgaben (Outputs) verfügen. Jeder In- und Output wird durch ein Schlüssel-Wert-Paar beschrieben, wobei der Schlüssel als `String` definiert ist und der Wert als Gleitkommazahl im Wertebereich zwischen $0,0$ und $1,0$. Bei boolschen Werten wird die $0,0$ als falsch und $1,0$ als wahr interpretiert.

Die in *Aroo* verwendeten Mappen haben zum Beispiel einen Input (Drückknopf mit dem Namen `"button"` (bool)) und einen Output (LED mit den Namen `"beacon"` (bool)). Die Konfiguration, welche In- und Outputs ein Gerät hat und welchen Namen diese haben, wird in einer XML-Datei gespeichert. Der Service lädt diese Datei beim Starten.

```
1 void OnManipulated(object sender,
2                    VIOServiceProxy.NodeEventArgs args)
3 {
4         // Die Eventargumente beinhaltet in args.node das
5         // auslösende Gerätund kann somit sofort angesprochen
6         // werden.
7         args.node.SetOutput("beacon", 1.0);
8 }
```

Code 3.8: Einfaches Beispiel für das Empfangen von Eingaben über VoodooIO und das Setzen von Ausgaben über die SetOutput-Methode.

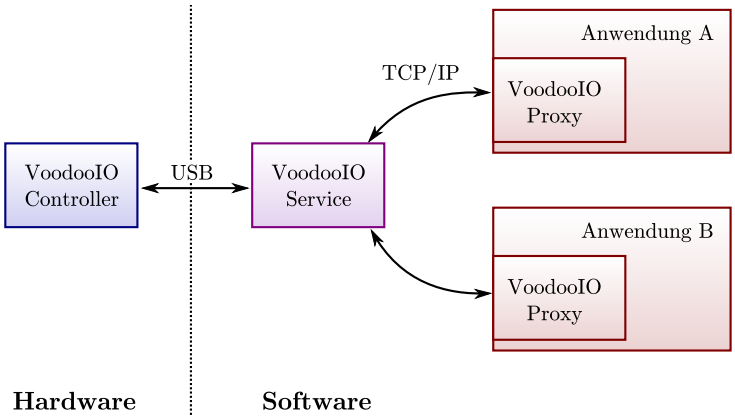

Abbildung 3.15: Aufbau der VoodooIO Kommunikation.

3.5.5 Tisch

Überblick

Die Verwendung verschiedener Technologien für Tracking, Interaktion und Darstellung soll am Tisch die Zusammenführung von virtuellen und realen Inhalten ermöglichen. Die Aufgabe der Tischanwendung ist es nun, diese Technologien sinnvoll zu kombinieren. Damit dies möglich wird, müssen zu einigen Technologien Schnittstellen geschaffen werden. Diese Schnittstellen und deren Implementierung werden im Folgenden beschrieben.

OneNote Anbindung

Der Zugriff auf Daten in OneNote erfolgt über ein in .NET entwickeltes *Application Programming Interface (API)*. Wie auch alle anderen .NET Bibliotheken kann diese in allen .NET-fähigen Sprachen [43] verwendet werden. Die Tischanwendung hingegen ist in ANSI C++ implementiert und kann nicht direkt auf solche Bibliotheken zugreifen. Bei dieser Konstellation ergeben sich folgende Probleme:

- *Managed vs. Unmanaged Code*: Microsofts .NET Framework basiert auf der *Common Language Infrastructure* (CLI) [9]. Dieses spezifiziert unter anderem einen architektur- und programmiersprachenunabhängigen lauffähigen Programmcode ähnlich dem Java Objectcode und deren Ausführung. Die Implementierung dieses Zwischencodes von Microsoft (MSIL) kann über die so genannte *Common Language Runtime* (CLR) ausgeführt werden. Die Speicherverwaltung eines solchen Codes wird von dieser Laufzeitumgebung automatisiert. Dieser Code wird daher allgemein als *managed Code* bezeichnet. Native Anwendungen, wie sie z.B. durch ANSI C++ Compiler erzeugt werden können,

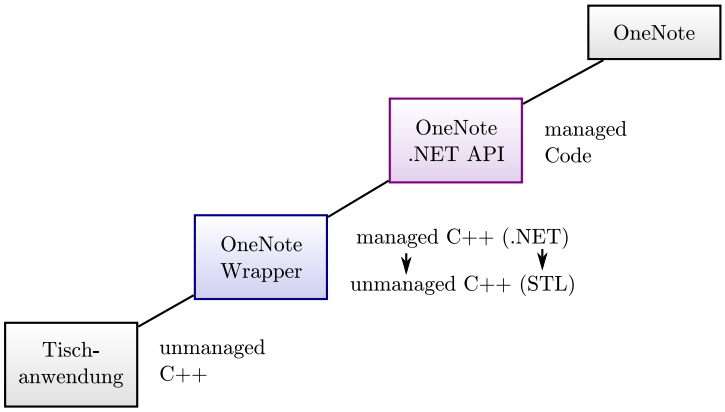

Abbildung 3.16: Der OneNote-Wrapper kapselt den Zugriff auf die .NET API von OneNote.

werden hingegen als *unmanaged Code* bezeichnet. Um nun in einer Anwendung Daten aus einem automatisch verwaltetem Speicherbereich verwenden zu können, müssen die Daten aus diesem extrahiert werden. Das kann nur mit einer Programmiersprache erfolgen, die mit beiden Modellen arbeiten kann. In unserem Fall ist das C++/CLI [8], eine von Microsoft für die Verwendung mit CLI erweitertes C++.

- *.NET vs. STL*: In C++ wird gerne mit der *Standard Template Library* (STL) gearbeitet, so auch in der Tischanwendung. Vor allem Klassen für Zeichenketten (`std::string`) und Vektoren (`std::vector`) werden gerne verwendet. Die in .NET verwendeten Klassen für diese Datentypen sind aber nicht kompatibel zur STL.

Damit nun das OneNote API von der C++ Anwendung für den Tisch verwendet werden kann, wurde ein eigener Konverter, ein so genannter *Wrapper*, entwickelt (siehe Abb. 3.16). Dieser hat zum einen die Aufgabe, managed Code vor der C++ Anwendung und umgekehrt unmanaged Code vor .NET zu verstecken. Zum anderen soll er implizit die Konvertierung von .NET in STL oder ANSI C Datentypen durchführen. Der Wrapper implementiert alle von der Tischanwendungen benötigten Methoden des OneNote APIs und stellt auch Hilfsfunktionen zur Verfügung, bei denen die Umsetzung in .NET einfacher ist als mit Standard C++ und STL.

DiamondTouch und Anoto Einbindung

Die Tischanwendung behandelt parallele Eingaben von diversen Geräten. Mehrere digitale Stifte können gleichzeitig in Einsatz sein, am DiamondTouch können bis zu vier Personen auf der Tischoberfläche interagieren [7]

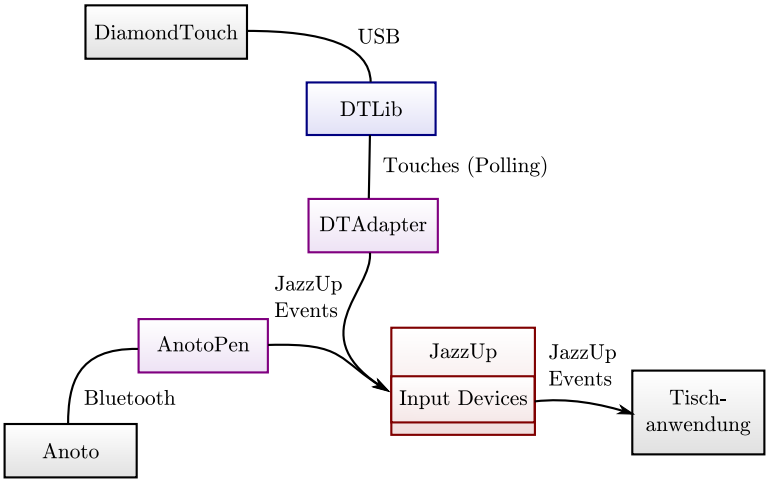

Abbildung 3.17: Aufbereitung der Eingaben vom DiamondTouch und den digitalen Stiften von Anoto.

und auch die Maus kann bewegt werden. Damit diese Eingaben von verschiedenen Geräten einfacher behandelt werden können, wird die Eingabegerätbehandlung der Bibliothek JazzUp verwendet. Hier werden Eingaben von verschiedenen Geräten in ein einziges Format konvertiert und können so von der Anwendung einheitlich empfangen werden. Damit JazzUp mit den verschiedenen Geräten arbeiten kann, muss für jedes Gerät eine Adapter implementiert werden, der die Rohdaten des Geräts in das von JazzUp verstehbare Format konvertiert. Dieser „Gerätetreiber“ muss hierfür das Interface `InputDevice` implementieren und von der Anwendung selbst instanziert und gestartet werden.

Für die Anbindung einer Maus und der digitalen Stifte gibt es bereits eine Implementierung. Die Bibliothek „AnotoPen“[17] liest die über Bluetooth übertragenen Events des digitalen Stifts aus und konvertiert sie in das passende Format. Die Anbindung an den DiamondTouch musste hingegen selbst entwickelt werden.

Im Diagramm 3.17 ist die Umgebung dieses Adapters (*DTAdapter*) dargestellt. Die DiamondTouch-SDK-Implementierung für C ist in die Bibliotheken *DTLib* und *DTIO* aufgeteilt [12]. DTIO ist für die systemnahe Kommunikation mit dem Systemtreiber zuständig. Das reicht von der maschinennahen Kommunikation mit der Hardware über USB bis zum Auslesen der von den Antennen empfangenen Werte. DTLib hingegen kapselt diese systemnahen Zugriffe und bietet für den weiteren Einsatz vorbereitete Daten an. Die von DTIO ausgelesen Werte der Antennen werden dabei ausgewertet und die „Bounding Box“ der Berührung berechnet. Weiters wird aus diesen

[17] http://www.officeoftomorrow.org

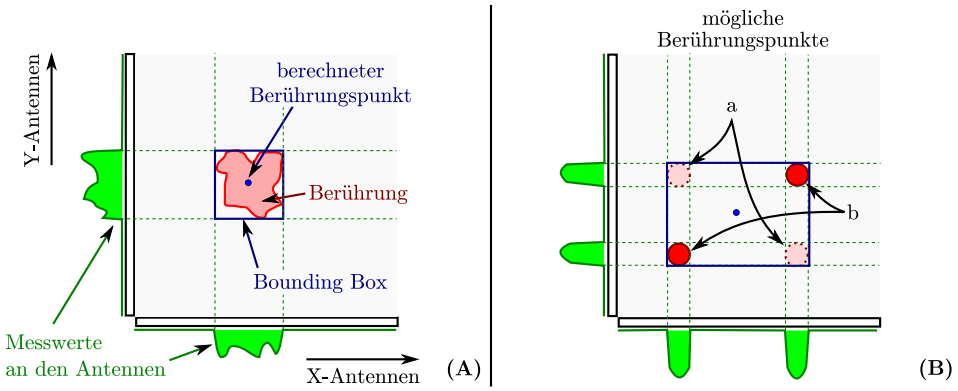

Abbildung 3.18: (A) Die Berechnung eines Berührungspunktes. Aufgrund der Detektionswerte der rechtwinklig zueinander stehenden Antennenreihen wird eine „Bounding Box“ berechnet, aus der dann ein Berührungspunkt berechnet wird. (B) DT kann mehrere Berührungspunkte von einer Personen nicht eindeutig erkennen, da mit den selben Antennenwerten mehrere verschiedene Ergebnisse berechnet werden können.

Informationen ein genauer Berührungspunkt berechnet. Pro Benutzer kann so eine „Bounding Box“ und ein Berührungspunkt eruiert werden. Zusätzlich kann über die DTLib auch die Kalibrierung des DiamondTouch ausgelesen und auf Berührungspunkte angewendet werden. Somit werden diese in das Bildschirmkoordinatensystem transferiert. In Abbildung 3.18 (A) sieht man wie aus den beiden im rechten Winkel zueinander liegenden Antennenreihen und deren Werte die „Bounding Box“ berechnet wird.

Der DTAdapter verwendet DTLib um auf diese Daten zugreifen zu können. Da DTLib nicht ereignisbasiert arbeitet, liest ein eigener Thread in regelmäßigem Abstand die Berührungspunkte und deren „Bounding Boxes“ aus und wandelt diese Daten in Events um.

Tracking der Mappen am Tisch

Das Tracking der Mappen am Tisch erfolgt über ARTag. Das Kamerabild dazu liefert die Klasse `Framegrabber` von JazzUp, der über DirectShow auf die Firewirekamera zugreift. Das Tracking läuft parallel zur Anwendung in einem eigenen Thread und generiert bei der Erkennung von Markern am Tisch eigene Events. Dabei wird zwischen den folgenden drei unterschieden:

- Marker wird zum ersten Mal erkannt
- vorher erkannter Marker ist verschwunden
- Marker wurde wieder erkannt

Diese Events werden vom Rendering (in der Klasse `FolderWidget`) verwendet, um zusätzliche Informationen an der Mappe am Tisch darzustellen.

ARTag berechnet die Markerposition und -orientierung im dreidimensionalen Raum. Auf der Tischfläche ist die Tiefeninformation und Skalierung aber nicht notwendig, da beide in diesem Setup konstant sind. Auch ist nur die Rotation um die auf die Tischoberfläche senkrecht stehende Koordinatenachse (Z-Achse) notwendig. Aufgrund dieser Vereinfachungen kann das Tracking stabilisiert werden. Da beide Marker an gegenüberliegenden Eckpunkten der Mappe angebracht sind, kann die Rotation auf dem Tisch relativ genau berechnet werden. Dafür wird der Unterschied des Winkels zwischen der Diagonale des Displays und der Strecke zwischen den beiden Markern kalkuliert. Die Entfernung zwischen den Markern ist für eine möglichst genaue Berechnung ein wichtiger Faktor. Da die Positionsbestimmung der einzelnen Marker abhängig von der Auflösung des Kamerabildes mit Fehlern behaftet ist, entstehen auch Fehler in der Rotationsberechnung. Durch einen größeren Abstand zwischen den Markern kann der Fehler aber verringert werden, da bei einem konstanten maximalen Fehler y und einem größer werdenden Abstand x der maximale Winkelfehler α bei $\alpha = atan(y/x)$ kleiner wird.

Diese Stabilisierung des Trackings funktioniert aber nur dann, wenn beide Marker erkannt werden können. Wird nur ein Marker erkannt, werden hingegen nur die X- und Y-Rotationswerte von ARTag ignoriert und keine sonstigen Stabilisierungen vorgenommen.

Zur Verringerung des „Jitterns“ des Trackings, werden die Werte zusätzlich noch geglättet, d.h. Position und Rotation werden über die arithmetische Mittelwertsberechnung einer definierbaren Anzahl (standardmäßig fünf) von Trackingergebnissen bestimmt. Somit ist die Bewegung der Augmentierung immer ein wenig verzögert. Da aber die Mappe die meiste Zeit still am Tisch liegt ist die Verzögerung kein großes Problem. Durch die Stabilisierung bewegt sich die Augmentierung im beschriebenen Setup und bei unbewegter Mappe maximal um nur einen Pixel.

Rendering

Für die Darstellung der virtuellen Inhalte werden zwei Bibliotheken eingesetzt (siehe Abb. 3.19):

- *wxWidgets*: Das Windowhandling, und somit auch die Eingabe über Maus und Tastatur, erfolgt über die mehrplattformfähige Bibliothek wxWidgets.

- *Widgets*: Das gesamte Rendering und auch die Verwaltung des Szenengrafen geschieht über die Bibliothek *Widgets*. Diese Bibliothek stellt sehr einfache, rotierbare GUI-Elemente, die mittels OpenGL dargestellt werden, zur Verfügung.

Da *Widgets* nicht sehr viele Arten von GUI-Komponenten unterstützt, wurden für *Aroo* noch Komponenten zur Augmentierung der Mappe und zum Ansteuern von Videos entwickelt.

- *FolderWidget*: Diese Komponente stellt die Augmentierung einer Mappe am Tisch dar. In Abbildung 3.20 ist die Funktionsweise dieser Komponente zu sehen. Über und rechts von der Mappe werden virtuelle Inhalte dargestellt, wobei in der oberen Leiste Bilder (mit roter Umrahmung) und Videos (mit gelber Umrahmung) angezeigt werden und in der seitlichen die Überschriften der einzelnen Seiten aus OneNote. Zusätzlich wird auf die Mappe selbst noch eine hellgraue Fläche projiziert, damit andere virtuelle Inhalte nicht auf die Mappe projiziert werden.

- *VideoWidget*: Das Darstellen von Videos wird von *Widgets* zwar unterstützt, aber diese Komponente hat keine Steuerungselemente zum Abspielen und Stoppen eines Videos. Aus diesem Grund wurde eine eigene Komponente zum Anzeigen von Videos entwickelt, die zusätzlich zwei Schaltflächen für Abspielen/Pause und Stopp besitzt.

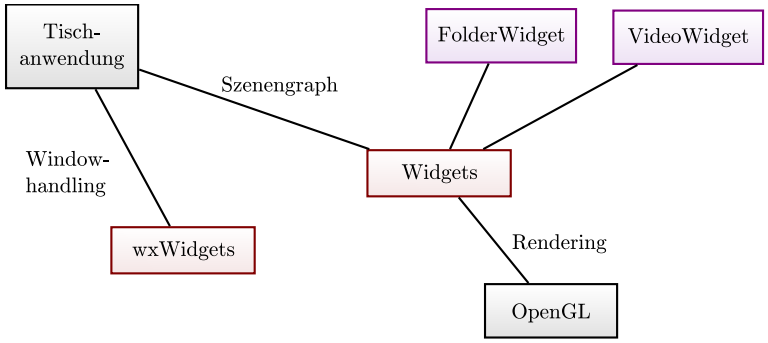

Abbildung 3.19: Aufbau des Renderings.

Datenhaltung

Die Verbindung von Daten und Darstellung wird von der Tischanwendung durch eine abgeänderte *Model-View-Controler* Architektur [14] realisiert, wobei der *Controler* in die *View* integriert ist (siehe Abb. 3.21). Jedes FolderWidget besitzt einen *Folder*, den es darstellt. Dieser Folder beinhaltet alle notwendigen Daten zum Mapping der verschiedenen IDs. Somit kennt jeder FolderViewer auch die ID der Marker für das Tracking auf dem Tisch. Jeder Folder besitzt weiters auch Dokumente, die von den verschiedenen Komponenten (ImageWidget, VideoWidget) im FolderWidget visualisiert werden.

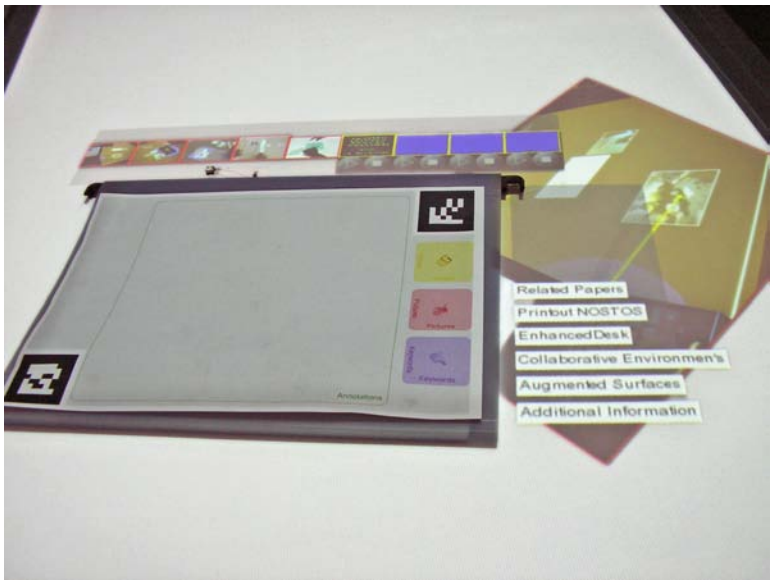

Abbildung 3.20: Die Augmentierung der Mappe wird von der Klasse `FolderWidget` erledigt. Videos und Bilder werden über der Mappe und die Seitentitel des zugehörigen OneNote Ordners rechts neben der Mappe angezeigt. Zusätzlich werden virtuelle Elemente auf der Mappe ausgeblendet.

Geladen werden die Daten aus OneNote über eine von `Folder` abgeleitete Klasse `OneNoteFolder`. Diese lädt beim Initialisieren alle notwendigen Dateien über den OneNote Wrapper und speichert diese als Objekte des Typs `Document` im `Folder` Objekt. Das FolderWidget selbst kümmert sich dabei nicht, woher die Daten kommen, die von ihm dargestellt werden. Somit können auch Daten aus anderen Quellen (z.B. Dateisystem oder andere DMS) dargestellt werden.

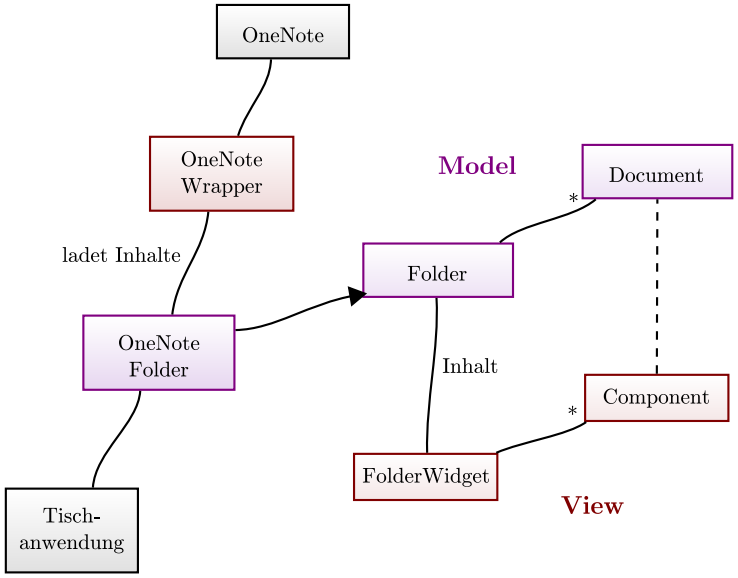

Abbildung 3.21: Aufbau der Datenhaltung: Über den OneNote Wrapper werden Ordner (Folder) und Dokumente (Document) geladen und von der View (FolderWidget und Component) dargestellt.

Kapitel 4

Resultate

Das Hauptziel der Anwendung war, den Vorgang des Suchens von realen Dokumenten zu beschleunigen und gleichzeitig die positiven Eigenschaften von Papier und realen Mappen, welche noch immer viele Vorteile gegenüber der digitalen Version haben [24], zu erhalten. *Aroo* stellt daher einige neue Interaktionsmöglichkeiten mit virtuellen Dokumenten am Computer als auch mit realen Dokumenten im Büro vor.

4.1 Navigation und Suche

Die Mappen und Ständer sind nach der Erweiterung um LED, Druckknopf und Drehknopf zur direkten Ein- und Ausgabe fähig. Sie bieten für Objekte, die üblicherweise keine Verbindung zu diesem haben, eine physikalische Schnittstelle zum Computer. Diese Ein- und Ausgaben sind zwar nur binär und daher in ihren Verwendungsmöglichkeiten beschränkt, doch für den Zweck der Navigation und des Durchsuchens von Ordnersystemen ausreichend. In Verbindung mit OneNote werden diese Erweiterungen in folgenden Hinsichten verwendet:

- bidirektionale Synchronisierung der Navigation
- erweiterte Suche

4.1.1 Bidirektionale Synchronisierung der Navigation

Die Navigation durch die Ordner in OneNote wird auch im Ständer visualisiert. Der aktuell ausgewählte und angezeigte Ordner wird durch das Leuchten des LEDs auf der zugehörigen Mappe angezeigt (siehe Abb. 4.1). Es ist somit sofort ersichtlich, wo die ausgedruckten Inhalte zum ausgewählten Ordner gefunden werden können. Der Inhalt der realen und virtuellen Ordner ist derselbe, aber die Reihenfolge der Mappen im Ständer kann völlig verschieden sein. Durch diese Hervorhebung der aktuell ausgewählten Mappe

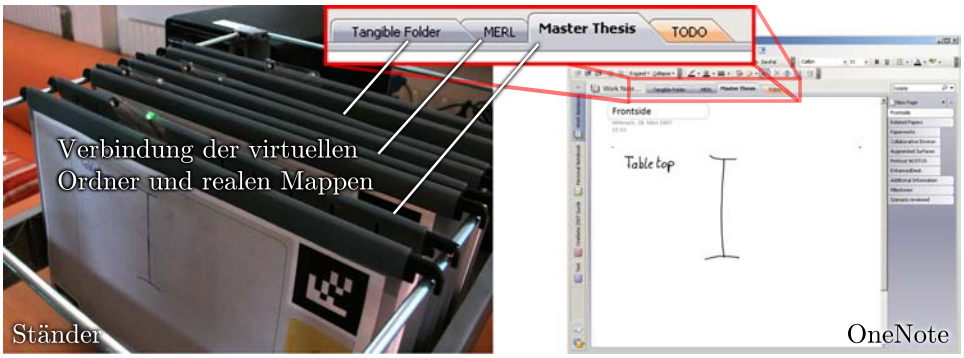

Abbildung 4.1: Synchronisation zwischen realen Mappen und virtuellen Ordnern: Die zugehörige Mappe zum aktuell angezeigten Ordner in OneNote wird durch das Leuchten des LEDs angezeigt.

ist die Verbindung zwischen den beiden Versionen aber sofort sichtbar. Das Ablegen von Dokumenten wird somit beschleunigt.

Die Navigation ist aber nicht nur einseitig, vom Computer aus, möglich, sondern funktioniert bidirektional. Über den Druckknopf auf der Mappe kann auch OneNote gesteuert werden. Drückt man diesen, so öffnet OneNote den zugehörigen virtuellen Ordner. Diese Funktion ist vor allem dann interessant, wenn ein Benutzer beim Durchsuchen der Mappen ein Dokument findet, zu dem er die digitalen Inhalte, wie z.B. Videos, sehen will.

4.1.2 Erweiterte Suche

Über Suchfunktion des OneNote Add-ins werden die Suchergebnisse nicht nur in OneNote angezeigt (Abb. 4.2), sondern auch an den Mappen. Dabei werden die LEDs aller Mappen, in derer virtueller Version Ergebnisse gefunden worden sind, aktiviert (siehe Abb. 4.3). Somit sieht man die Ergebnisse nicht nur am Bildschirm, sondern auch im Ständer. Da am Ständer die Anzahl der Suchergebnisse pro Mappe nicht dargestellt werden kann, kann über den Drehknopf am Ständer die Suchschärfe angepasst werden. Dreht man diesen Knopf, so wird der Schwellwert, ab welcher Anzahl von Ergebnissen pro Ordner das LED leuchten soll, verändert. So ist es möglich, die Ordner mit den meisten Suchtreffern zu finden.

4.2 Besprechungen

Die Erweiterungen des Besprechungstisches erlauben eine Integration virtueller Inhalte in die Diskussion. Dies geschieht, ohne dass die Situation am Tisch grundlegend verändert wird. Der Tisch kann noch immer als normaler Tisch, auf den man zum Beispiel eine Kaffeetasse stellen oder seine Unter-

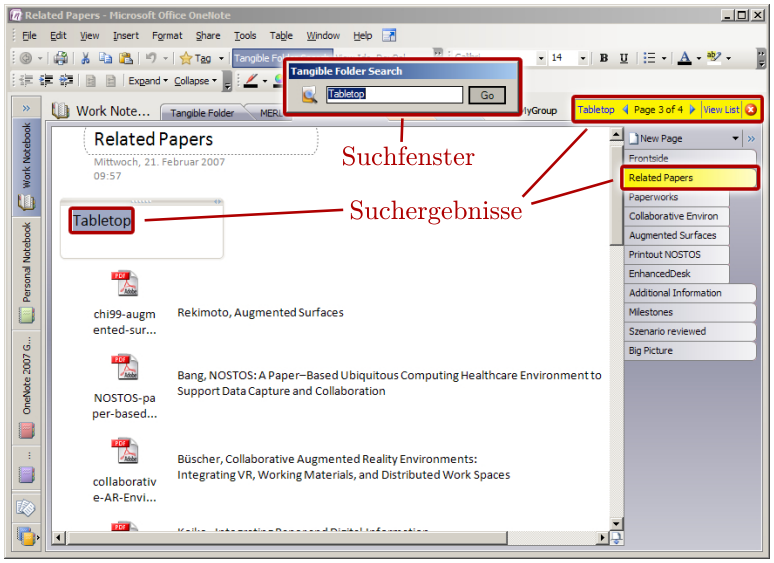

Abbildung 4.2: Die erweiterte Suche wird durch ein zusätzliches Suchfenster ermöglicht.

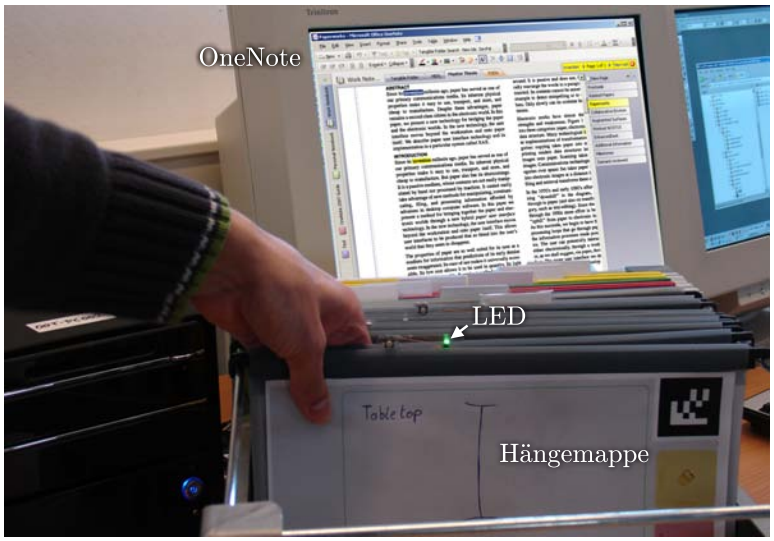

Abbildung 4.3: Die Suchergebnisse werden sowohl in OneNote als auch an den Mappen visualisiert.

Abbildung 4.4: Eine Diskussion am Besprechungstisch bei der virtuelle wie reale Dokumente nebeneinander verwendet werden.

lagen legen kann, verwendet werden. Die Projektion und das Tracking am Tisch erlauben aber zusätzlich die Verwendung von virtuellen Inhalten. Wie in Abbildung 4.4 zu erkennen ist, können so neben ausgedruckten Dokumenten auch Videos und Bilder gelegt werden.

Die Interaktion mit diesen virtuellen Inhalten geschieht mit den Fingern und es ist kein zusätzliches Eingabegerät notwendig. Mit seinem Finger kann der Benutzer virtuelle Objekte mittels einer integrierten Rotation und Bewegung [21,29] anordnen oder durch das Auseinander- oder Zusammenbewegen von zwei Fingern skalieren [6].

Das Laden von virtuellen Objekten geschieht über die Mappe. Legt der Benutzer die Mappe auf den Tisch werden alle virtuellen Inhalte des verknüpften virtuellen Ordners neben diesen dargestellt. Diese können einfach mit den Fingern „aus“ der Mappe gezogen werden. Die virtuellen Inhalte (alle in den virtuellen Ordner eingefügten Bilder und Videos) werden dabei aus OneNote geladen.

Da in manchen Ordnern eine sehr große Anzahl von virtuellen Inhalten gespeichert sein kann, können diese mit dem digitalen Stift zusätzlich gefiltert werden. Tippt der Benutzer auf eine der Schaltflächen „Videos“, „Pictures“ oder „Keywords“ auf der Vorderseite der Mappe, so werden die entsprechenden virtuellen Informationen angezeigt oder ausgeblendet.

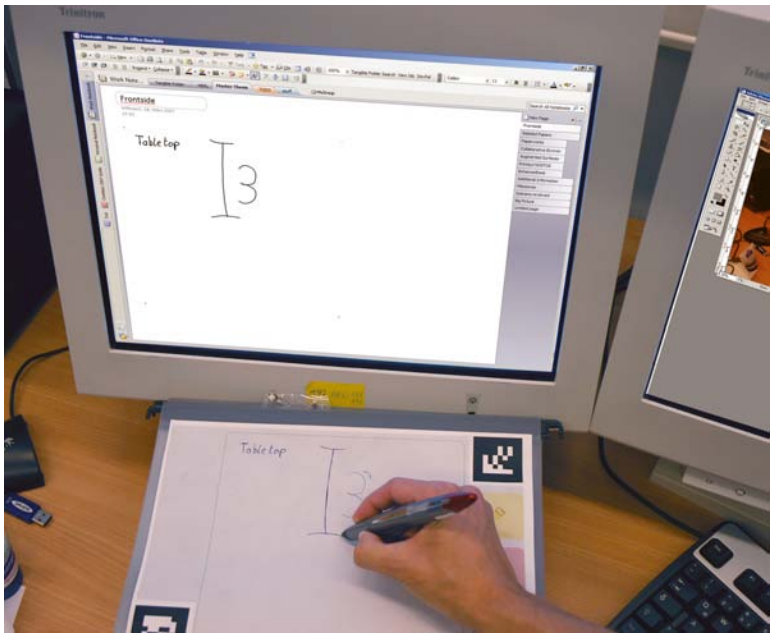

Abbildung 4.5: Notizen, die auf der Vorderseite der Mappe mit dem digitalen Stift geschrieben werden, sind auch in OneNote verfügbar.

4.3 Notizen auf der Mappe

Zur besseren Differenzierbarkeit der Mappen und zur Beschreibung des Inhalts kann die Vorderseite der Mappen mit einem digitalen Stift beschriftet werden. Diese Beschriftung wird dabei nicht nur physikalisch gespeichert, sondern auch virtuell (siehe Abb.4.5). Die Striche werden vom digitalen Stift in OneNote übertragen und dort auf die erste Seite des virtuellen Ordners als eine Skizze eingefügt. Diese Informationen können in weiterer Folge in OneNote zu Handschrift umgewandelt, mit der integrierten Schrifterkennung als Text interpretiert und somit ebenfalls durchsucht werden. So sind selbst Notizen an der Mappe digital verfügbar.

Kapitel 5

Diskussion

Obwohl noch eine formale Benutzerstudie ausständig ist, gibt es schon Ergebnisse in Hinblick auf die Benutzbarkeit. Diese wurden aus informellen Rückmeldungen und eigenen Erfahrungen gewonnen.

5.1 Ablegen von Dokumenten

Das Ablegen von Dokumenten wird durch die bidirektionale Navigation beschleunigt. Legt ein Benutzer ein Textdokument in einem Ordner in OneNote ab, so muss er vorher auf eine gewünschte Seite navigieren. Dadurch wird auch gleichzeitig die zugehörige Mappe durch das Leuchten des LEDs hervorgehoben. Wird in Folge dieses Dokument ausgedruckt, so braucht beim Ablegen nur das leuchtende LED gesucht werden um die richtige Mappe zu finden. Die Hervorhebung der verbundenen Mappe beschleunigt dabei das Ablegen merkbar, da diese so viel schneller gefunden werden kann.

5.2 Darstellung virtueller Inhalte

Mit dem Tabletopdisplay und durch das Tracking der Mappen am Tisch können virtuelle Inhalte sehr leicht und angenehm dargestellt werden. Die Verwendung der Mappe als eine Art Datenspeicher und *Tangible Interface* [16] ermöglicht, dass am Tisch weder Maus noch Tastatur verwendet werden müssen. Gleichzeitig erkennt der Benutzer sofort, welche virtuellen Inhalte im verknüpften Ordner gespeichert sind, da am Tisch die Bilder und Videos direkt neben der Mappe angezeigt werden.

Zusätzlich ist die Organisation virtueller Inhalte einfach zu vollziehen. Sie können in OneNote einfach per Drag'n'Drop auf eine Seite gezogen werden und sind somit schon bei der nächsten Verwendung der verknüpften Mappe am Tisch verfügbar.

5.3 Besprechungsvorbereitung

Da mit OneNote die Zusammenstellung von Ordnern einfach ist, kann eine Besprechung gut vorbereitet werden. Man nimmt eine leere Mappe und navigiert mit dem Druckknopf auf der Mappe zum verbundenen OneNote-Ordner. In diesen kopiert und schreibt man die notwendigen Dokumente und Informationen. Gleichzeitig können alle notwendigen Informationen ausdruckt und in die Mappe ablegt werden. Zur besseren Übersichtlichkeit kann nun die Vorderseite der Mappe zusätzlich beschriftet werden. Man hat somit einen Ordner, der sowohl virtuell als auch real alle notwendigen Daten beinhaltet und braucht während der Besprechung am Besprechungstisch nur eine einzige Mappe.

5.4 Interaktion am Tisch

Durch die Verwendung des DT am Besprechungstisch ist eine intuitive Interaktion mit virtuellen Inhalten möglich. Dass Videos und Bilder mit den Fingern ähnlich bewegt werden wie reale Dokumente, ermöglicht eine sehr kurze Lernphase. Da die integrierte Bewegung und Rotation von Objekten für die Benutzer ähnlich wie eine physikalisch korrekte Bewegung wirkt, wird diese Interaktion schnell verstanden. Auch das Skalieren mit zwei Fingern benötigt außer der Anmerkung, dass es diese Interaktion gibt, keiner genaueren Erläuterung.

5.5 Suche

Die integrierte Suche von digitalen und realen Dokumenten liefert gute Ergebnisse. Reale Dokumente können ein wenig schneller gefunden werden, da die Suche zumindest auf einzelne Mappen eingeschränkt wird. Im Testsetup war die Anzahl der Mappen (vier) aber viel zu gering, um eine signifikante Zeitersparnis festzustellen. Erst bei einer größeren Anzahl von Mappen und Ständern sollte eine Beschleunigung der Suche zu erkennen sein.

5.6 Anwendungsspezifische Einschränkungen

Während der Entwicklung von *Aroo* wurden einige Probleme und Einschränkungen sichtbar, die zum einen mit den Technologieentscheidungen zusammenhängen und zum anderen bei der Verwendung aufgefallen sind.

5.6.1 Komplexität virtueller DMS

Verglichen zu virtuellen DMS sind physikalische DMS in der Komplexität des Aufbaus eingeschränkt. Zum einen können virtuelle DMS beliebig tiefe Hier-

archien aufweisen, denn anders als reale Ordnersysteme ist auch das Ablegen von Ordnern in andere Ordner (usw.) möglich. Zum anderen verwenden virtuelle DMS nicht nur strikte Hierarchien, sondern auch Querverweise oder netzwerkartige Strukturen, die in der physikalischen Welt schwer oder gar nicht nachzubilden sind. Bei der Implementierung solcher Systeme muss in Kauf genommen werden, dass die physikalische und virtuelle Struktur nicht dieselbe ist.

5.6.2 Falsche Einordnung

Aroo geht von der Annahme aus, dass die Inhalte in beiden Versionen der verknüpften Ordner gleich sind. Es besteht aber die Möglichkeit, dass beim Ablegen von Dokumenten Fehler passieren und diese in eine andere Mappe gegeben werden. In diesem Fall gibt es keine Unterstützung vom System, dieses Dokument wieder zu finden.

Dieses Problem würde sich durch die Erkennung der einzelnen Dokumente lösen lassen. Würde nicht nur die Mappe erkannt, sondern die Dokumente in der Mappe selbst, so könnte dem Benutzer zum Beispile eine Warnung angezeigt werden. Alternativ könnte das System diese „Falscheinordnung“ abspeichern und diese bei Suchanfragen beachten.

5.6.3 Wartung

Die Wartung der physikalischen Mappen wurde im Szenario nicht betrachtet. Um *Aroo* in einer realen Umgebung verwenden zu können, fehlt es noch ein Werkzeug, das das Erzeugen von Mappen vereinfacht. Es muss darauf geachtet werden, dass jede Mappe eindeutig identifiziert werden kann. Das bedeutet, dass Anotopapier, ARTag-Marker und VoodooIO-Geräte-ID immer eindeutig sein müssen.

Es fehlt weiters noch eine Anwendung, die neue Mappen mit virtuellen Ordnern verbindet. Dabei ist die Speicherung des Mappings in einer separaten XML-Datei nicht optimal, da diese Informationen auch direkt in den Metadaten der OneNote-Ordner gespeichert werden könnte.

5.7 Technologische Probleme

5.7.1 OneNote

Obwohl OneNote 2007 ein gutes API zum Zugriff auf Daten verfügt, wurden doch einige Probleme entdeckt.

- *Behandlung von Video- und Audiodateien:* OneNote neigt beim Aufnehmen von Videos zu Abstürzen. Bisher wurde aber noch kein Fall von Datenverlust beobachtet. Ein weiteres Problem ist die Anzeige von

Videos, die nicht direkt auf einer Seite von OneNote integriert ist, sondern in einem separaten Fenster, dass über eine Toolbar von OneNote gesteuert wird. Das Videoausgabefenster selbst hat keine Schaltflächen zur Steuerung. Weiters können externe Videos und Audiodateien nicht als `MediaFiles` in OneNote gespeichert werden. Solche Dateien können nur mit einem externen Programm geöffnet werden, auch wenn OneNote das Abspielen dieser Videos unterstützen würde (z.B. bei Windows Media Dateien).

- *Add-in Schnittstelle:* Mit einem Add-in ist es nicht möglich, eigene Menüpunkte, Toolbars oder ein anderes Widget als einen Button zu erzeugen. Somit muss die Benutzerschnittstelle in ein eigenes Fenster ausgelagert werden. Ein solches Fenster erscheint beim Erzeugen in einem Eventhandler aber meistens im Hintergrund und ist standardmäßig modal. Das führt dazu, dass die Benutzeroberfläche von OneNote nicht mehr reagiert und der Benutzer den Eindruck eines abgestürzten OneNotes erhält. Ein weiteres Problem ist, das das Navigations-Event von OneNote erst nach ca. einer Sekunde Stillstand der Maus an die Add-ins weitergeleitet wird. Daher reagieren die LEDs an den Mappen erst nach einer Verzögerung.

- *Unzureichende Dokumentation:* Die verfügbare Dokumentation des OneNote APIs ist sehr kurz und wenig ausführlich. Es sind zwar alle Methoden und die XML-Struktur beschrieben, doch geht diese nicht in die Tiefe. Zusätzlich ist die einzige verfügbare Dokumentation auf dem Stand der Beta-Version, doch seither haben sich einige Funktionen in deren Parametern verändert. Die XML Struktur ist derzeit leider ebenfalls so schlecht beschrieben, dass bei vielen Attributen nicht genau gesagt werden kann, welche Aufgabe sie besitzen und wie sie eingesetzt werden.

5.7.2 DiamondTouch

DiamondTouch ist sehr gut als Trackingoberfläche für Mehrbenutzeranwendungen geeignet, da die Berührungspunkte verschiedener Benutzer unterschieden werden können. Ein Problem ist aber, dass der DiamondTouch *keine vollständige Mehrpunktinteraktion* erlaubt. Es ist zwar möglich, die Anzahl der Berührungspunkte am Tisch in den meisten Fällen zu bestimmen, doch nicht ihre Position. Bei zwei Berührungspunkten gibt es immer zwei Varianten, wo die Oberfläche berührt worden ist. Somit ist keine echte Mehrpunktinteraktion möglich, sondern nur eine Interaktion mittels „Bounding Box“ und Anzahl der Berührungspunkte.

5.7.3 Tracking

Obwohl ARTag besser mit unterschiedlichen Lichtverhältnissen zurecht kommt als ARToolkit [18], gibt es bei extremen Beleuchtungsunterschieden auf der Trackingfläche Probleme. Diese rühren von der Kamera, da der Dynamikumfang dieser begrenzt ist. Problematisch ist dies besonders bei einer kontrastreichen Projektion und schwachem Umgebungslicht. Marker in dunklen oder hellen Bereichen der Projektion werden hier nicht erkannt, da sie im Kamerabild nicht sichtbar sind.

Eine Lösung dieses Problems wäre der Einsatz einer Infrarotkamera, da das projizierte Bild beim LCD Projektoren im Infrarotbereich einheitlich intensiv ist, d.h. das Bild nicht sichtbar ist.

5.7.4 Digitale Stifte

Die Bibliothek AnotoPen funktioniert derzeit nur mit dem Windows Bluetooth Stack. Die meisten Hersteller für Bluetooth Empfänger liefern aber nur einen Treiber für den Widcomm Bluetooth Stack[1] mit und nicht für alle Empfänger gibt es Treiber für den Microsoft Stack [28]. Eine Bibliothek, die mit beiden Stacks funktioniert, ist nach der Implementierung dieser Arbeit fertiggestellt, aber nicht mehr eingebaut worden.

5.7.5 Widget Bibliothek

Diese (noch in Entwicklung befindliche) Bibliothek unterstützt zwar die Entwicklung von rotierbaren GUI Elementen, doch ist die Funktionalität auf das Rendering, Verwalten von GUI-Elementen und ein paar wenige Interaktionsmöglichkeiten eingeschränkt. Selbst Schaltflächen und die gesamte Abarbeitung von Eingaben mussten zusätzlich entwickelt werden.

5.7.6 VoodooIO

VoodooIO[2] befindet sich zur Zeit des Verfassens dieser Arbeit noch in Entwicklung. Daher ist der Proxy noch nicht ausgereift und ist noch nicht vollständig implementiert. Ein Problem war, dass das Setzen von Outputs nicht zu schnell hintereinander geschehen durfte, da der Befehl ansonsten ignoriert wurde. Erst mit einem zeitlichen Abstand von 200 ms zwischen zwei Befehlen reagierten die Geräte korrekt.

Ein weiteres Problem war der Drehknopf am Ständer, der selbst bei Nichtbetätigung Events schickte. Besonders problematisch hierbei war, dass dieser dabei eine Wertveränderung von bis zu ± 0.25 (bei einem Wertebereich von 0.0 bis 1.0!) aufwies. Somit waren die Ergebnisse beim Einschränken der Suchergebnisse am Ständer nicht zufriendenstellend.

[1] http://www.broadcom.com/products/bluetooth_sdk.php
[2] http://www.voodooio.com

Kapitel 6

Verwandte Arbeiten

Es wurden in der Vergangenheit schon einige Methoden und Systeme vorgestellt, die das Schließen der Lücke zwischen virtuellen und realen Dokumenten thematisierten.

6.1 Augmentierung von Papier

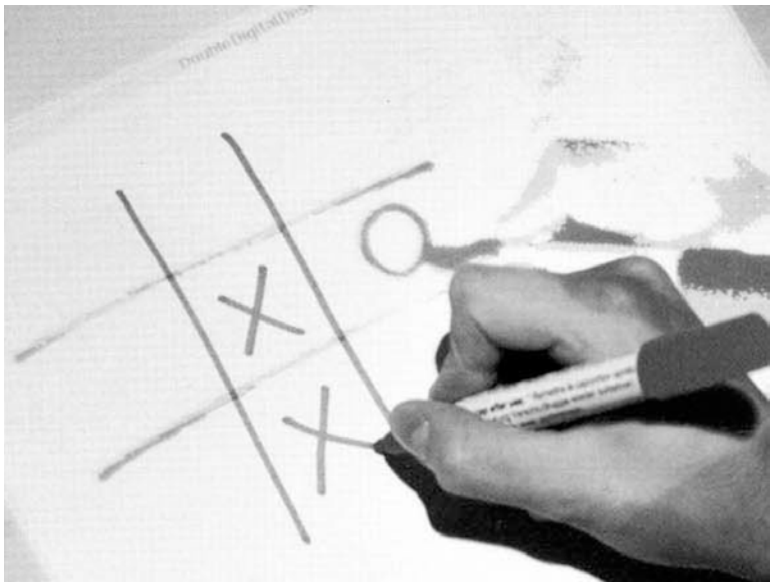

Abbildung 6.1: Digital Desk: Augmentierung der Tic-Tac-Toe Spielfläche um die gegnerischen Züge [42].

Einer der ersten Ansätze der Erweiterung von Papier durch virtuelle Inhalte war der *Digital Desk* [42]. Auf einem Tabletop-Display wurde Papier durch eine Projektion augmentiert. Dabei wurde durch die Projektion auf

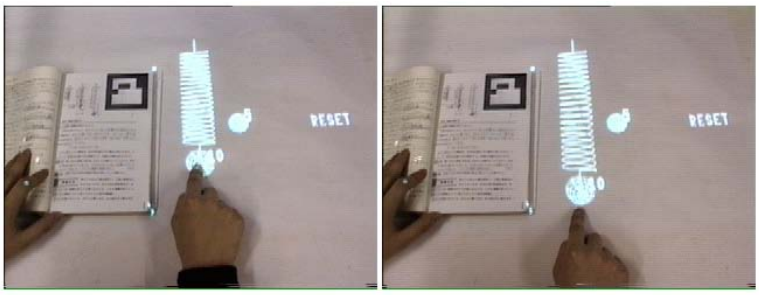

Abbildung 6.2: EnhancedDesk: Virtuelle, interaktive Inhalte werden neben das Buch projiziert. Mit den Fingern kann mit diesen interagiert werden [20].

Papier versucht, gemeinsam auf einem Blatt Papier zu zeichnen. Das Papier und ein normaler Stift wurden hierbei auch als Eingabegerät verwendet, indem das Geschriebene von der Kamera erfasst und analysiert worden ist. Eine Ausprägung davon ist ein Tic-Tac-Toa-Spiel (siehe Abb.6.1), wobei die eigenen Spielzüge einfach auf Papier gezeichnet, von einer Kamera aufgezeichnet und die des Gegners auf die Spielfläche projiziert wurden.

Mit der Erweiterung des Digital Desk, dem *Enhanced Desk* [19], wurde die Idee des augmentierten Papiers noch weiter entwickelt. Die Idee hierbei war die Erweiterung von Büchern durch interaktive Animationen. Dazu wurde auf die Seiten von Büchern Marker gegeben über die virtuelle Inhalte gestartet wurden (siehe Abb. 6.3). Diese wurden neben das Buch projiziert und mit den Fingern war eine Interaktion möglich.

6.2 Speicherung über Platzhalter

Rekimoto et. al. [32] zeigte mit „Augmented Surfaces“ wie digitale Daten mit einem realen Objekt verbunden werden können (Abb. 6.3). Über Drag'n'Drop werden auf einem Tisch Bilder oder Videos auf mit Markern getrackte Gegenstände (z.B. eine Videokassette) gezogen und somit darauf gespeichert. Zum Laden werden diese Objekte einfach wieder auf den Tisch gelegt und die digitalen Inhalte angezeigt.

Eine ähnliche Verbindung von digitalen Inhalten mit realen Gegenständen wurde in „mediaBlocks„ erreicht [36]. Im Unterschied zu „Augmented Surfaces“ wurden hier aber abstrakte Blöcke verwendet, die in der Realität selbst keine Verwendung haben.

6.3 Integration von Papier in DMS

Im Bereich von DMS integrierten *Xerox PaperWorks* [17] und *Portfoil* [31] reale Ausdrücke in den Arbeitsablauf eines großen elektronischen DMS. Zu

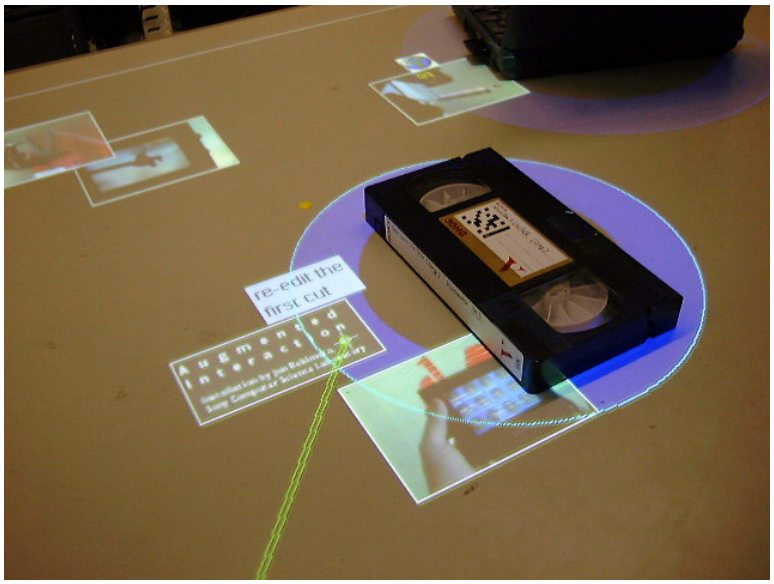

Abbildung 6.3: Bei Augmented Surfaces können Dokumente per Drag'n'Drop auf mit Markern getrackten realen Objekten gespeichert werden [32].

Dokumenten wurde hier eine Art Benutzeroberfläche auf Papier gegeben, auf das Beschreibungsdaten und Verarbeitungsbefehle geschrieben werden konnten. Die Abarbeitung der geschriebenen Information geschah hier beim Einscannen in das System. Zusätzlich wurde beim Ausdrucken von Dokumenten jedem eine elektronisch lesbare Kennung hinzugefügt, damit auf die elektronische Version schneller zugegriffen werden konnte.

Arregui et. al. [2] versuchte, wie auch in dieser Arbeit, die Organisation von Ausdrucken mit digitalen DMS zu verbinden. Jeder Ausdruck wird beim Ausdruck mit einem RFID Sender ausgestattet und jeder Ordner mit RFID Lesegeräten. Beim Ausdrucken wird das reale Dokument automatisch mit der digitalen Version verbunden. Über die Lesegeräte in den Ordnern kann jedes Dokument geortet werden und der Aufbewahrungsort am Computer angezeigt werden. Anders aber als *Aroo* besitzen die realen Ordner selbst keine direkte Ein- und Ausgabemöglichkeit.

6.4 Verknüpfung von virtuellen und realen Dokumenten

Da Ausdrucke aber nicht nur gesammelt und geordnet werden, sondern mit ihnen auch gearbeitet wird, ist es auch notwendig jede einzelne Seite direkt mit digitalen Daten zu verknüpfen.

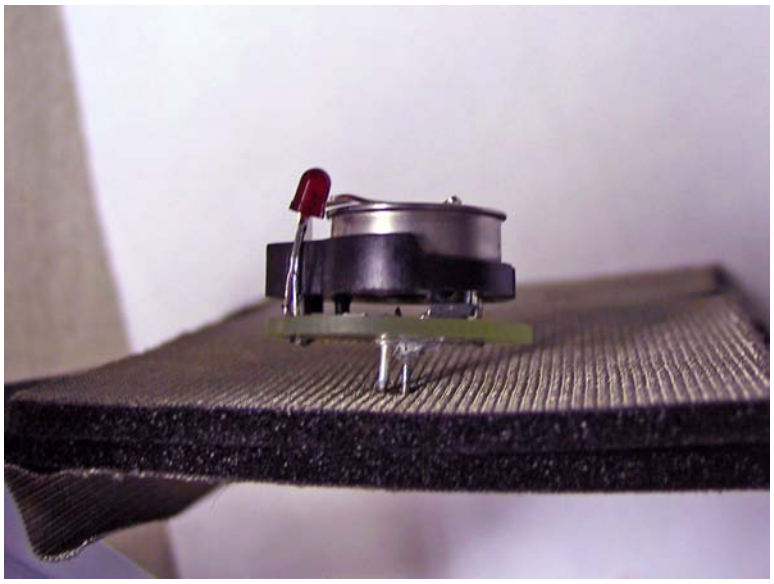

Abbildung 6.4: VoodooIO: Einfache Geräte können beliebig in die Pin&Play-Oberfläche gesteckt werden [38].

Paper Augmented Digital Documents [15] ermöglichen das Bearbeiten von Dokumenten am Computerbildschirm wie am Papier. Dokumente werden mit einer eindeutigen ID versehen auf Papier gedruckt und können mit digitalen Stiften bearbeitet werden. Hierbei wird die Anoto-Technologie verwendet, die es auch ermöglicht jedes Blatt Papier zu identifizieren. Die mit dem Stift gezeichneten Striche sind natürlich am Ausdruck sichtbar, werden aber gleichzeitig in das digitale Dokument eingefügt. Somit ist kein Einscannen notwendig und auch der Medienbruch wird verhindert.

6.5 Pin&Play / VoodooIO

VoodooIO (auch bekannt unter den Namen *Pin&Play*) [39] wurde entwickelt um einfache Geräte (Schalter, Drehknöpfe, usw.) vernetzen zu können. Diese Geräte sind dabei aber nicht fixiert, so wie zum Beispiel verkabelte Geräte, sondern können einfach aus dem Netzwerk genommen und an einer anderen Stelle wieder eingefügt werden. Die Netzwerkkommunikation passiert hierbei über zwei Leiter, mit denen die Geräte verbunden sind. In der originalen Verwendungsweise werden diese Geräte auf eine Oberfläche gesteckt, in die zwei übereinanderliegende Drahtgeflechte eingebaut sind (siehe Abb. 6.4). Statt dem Drahtgeflecht können aber auch andere leitende Materialien verwendet werden.

Kapitel 7

Resumé und Ausblick

Mit *Aroo* wird versucht die Trennung zwischen der Organisation von digitalen und realen Dokumenten zu verringern. Dabei war es von Wichtigkeit, nicht nur ein einzelnes Detail dieser Problematik zu betrachten, sondern ein ganzes Szenario vom Erstellen von Dokumenten über deren Benützung an verschiedenen Orten bis zu ihrer Organisation abzudecken.

Ein besonderes Anliegen war es auch, virtuelle Inhalte in Diskussionen in einer angenehmen Art und Weise einzubinden. Über den Einsatz von Hängemappen und dem DiamondTouch wurde eine Möglichkeit zum Laden von virtuellen Inhalten auf einer Tabletop-Anwendung geschaffen. Normalerweise besteht auf solchen Mehrbenutzerarbeitsflächen das Problem, dass keine Tastatur und Maus verfügbar sind und somit die gängigen Methoden zum Laden schwer anwendbar sind. Auch werden solche Methoden am Tisch manchmal als störend empfunden, besonders wenn der Tisch nicht als Computerbildschirm wahrgenommen wird, sondern nur als Erweiterung des Tisches.

In Hinblick der Weiterentwicklung von *Aroo* gibt es zwei Ansatzpunkte:

- Der erste ist die Erweiterung der Verbindung zwischen virtuellen und realen Objekten. So wäre auch die Verbindung von einzelnen Seiten der Ausdrucke mit deren digitalen Version interessant. Dabei sollte nicht nur auf die Möglichkeit zum Bearbeiten dieser Seiten [24] ein Auge geworfen werden, sondern auch auf die Erkennung der Ausdrucke in der Mappe selbst [2] und welche Auswirkung dies auf das virtuelle DMS haben könnte.

- Ein wichtiger Punkt ist auch eine Evaluierung des Systems. Alle bis jetzt erlangten Ergebnisse sind aus informellen Rückmeldungen gewonnen worden. Eine formelle Studie wäre notwendig, um sagen zu können, ob ein solches System für den Einsatz in einem Büro wirklich geeignet ist oder welche Teile noch verbessert werden müssten. Auch sollte diese Benutzerstudie mit einer größeren Anzahl von Map-

pen durchgeführt werden, da die aktuelle Anzahl nicht repräsentativ ist.

Papier wird in der nahen Zukunft wahrscheinlich nicht durch andere Medien ersetzt werden. Somit werden auch Hängeregistraturen und Aktenschränke nicht sehr bald aus den Büros verschwinden. Die Verbindung von Papier und Ordner zu deren virtuellen Versionen und die Erweiterung von Bürogegenstände um Ein- und Ausgabegeräte können die Arbeit im Büro trotzdem erleichtern. Durch die Erweiterung allseits bekannter Gegenstände ändern sich die wesentlichen Eigenschaften dieser kaum. Die Vorteile bleiben bestehen, aber manche Nachteile können beseitigt werden.

Anhang A

Addin Quellcode

A.1 AddinSample.cs

```
1 using OneNoteAddIn;
2 using System.Runtime.InteropServices;
3 using OneNote = Microsoft.Office.Interop.OneNote;
4 using System.Xml;
5 using System.Diagnostics;
6 using VoodooIO;
7 using System;
8
9 namespace OneNoteSampleAddin
10 {
11     [Guid("87E61590-99E0-46d3-99F1-15B95092CF13")]
12     public class AddInSample : IOneNoteAddIn
13     {
14
15
16         // VoodooIO service connection
17         const string vioHost = "localhost";
18         const int vioPort = 5555;
19
20         // VoodooIO Proxy
21         private VoodooIO.VIOServiceProxy proxy;
22
23         // OneNote XML Namespace
24         const string strNamespace = "http://schemas.microsoft.
               com/office/onenote/2007/onenote";
25
26         // OneNote API
27         private OneNote.Application onApp;
28
29         public AddInSample()
30         {
31             // Startup OneNote API
32             onApp = new OneNote.Application();
33
```

```
34              // Create Proxy object
35              proxy = new VIOServiceProxy("SampleAddIn");
36              // Connect to VoodooIO service
37              proxy.Connect(vioHost, vioPort);
38              // Connect all events
39              proxy.NodeManipulated += new VIOServiceProxy.
                    NodeManipulatedHandler(proxy_NodeManipulated);
40              proxy.NodeAdded += new VIOServiceProxy.
                    NodeAddedHandler(proxy_NodeAdded);
41              proxy.NodeRemoved += new VIOServiceProxy.
                    NodeRemovedHandler(proxy_NodeRemoved);
42          }
43
44          /***************************/
45          /* VoodooIO Event Handler: */
46          /***************************/
47
48          void proxy_NodeRemoved(object sender, VIOServiceProxy.
                NodeEventArgs args)
49          {
50              Trace.Write("Tangible␣Folder␣Search:␣Node␣" + args.
                    node.Id + "␣removed");
51          }
52
53          void proxy_NodeAdded(object sender, VIOServiceProxy.
                NodeEventArgs args)
54          {
55              Trace.Write("Tangible␣Folder␣Search:␣Node␣" + args.
                    node.Id + "␣added");
56          }
57
58          void proxy_NodeManipulated(object sender,
                VIOServiceProxy.NodeEventArgs args)
59          {
60              // activate LED on the folder which sent the event.
61              args.node.SetOutput("beacon", 1.0);
62          }
63
64          /*****************************/
65          /* OneNote API Event Handler */
66          /*****************************/
67
68          public bool OnClick([In] string strActivePageID)
69          {
70              MessageBox.Show("Currently␣viewed␣Section␣ID:␣" +
                    getCurrentSectionId());
71              return true;
72          }
73
74          public bool OnEvent([In] OneNote.OneNoteAddIn_Event evt
                , [In] string strParameter)
75          {
76
77              // Handle the two different events:
```

```
78              switch (evt.ToString())
79              {
80                  case "evtAddInHierarchyChange":
81                      // Some data has changed
82                      // do nothing
83                      break;
84                  case "evtAddInNavigation":
85                      MessageBox.Show("You␣have␣navigated␣to␣the␣
                            page␣(id):␣" + strParameter);
86                      break;
87              }
88              return true;
89          }
90
91
92          /******************/
93          /* Sample Methods */
94          /******************/
95
96          public string getCurrentSectionId(){
97              // Get hierarchy data from root to the pages
98              string strXML;
99              onApp.GetHierarchy(null, // from root
100                                 OneNote.HierarchyScope.hsPages,
                                       // to pages
101                                out strXML); // generated XML-
                                       Text
102
103             // Load XML-Text into the .NET XML parser
104             XmlDocument xmlDoc = new XmlDocument();
105             xmlDoc.LoadXml(strXML);
106
107             // Create an XmlNamespaceManager for resolving
                    namespaces.
108             XmlNamespaceManager nsmgr = new XmlNamespaceManager
                    (xmlDoc.NameTable);
109             nsmgr.AddNamespace("one", strNamespace);
110
111             // Query the actual viewed Section by using XPath.
112             string xpQuery = "//one:Section[@isCurrentlyViewed
                    =\"true\"]";
113             XmlElement xmlActiveSection =
114                (XmlElement)xmlDoc.SelectSingleNode(xpQuery,
                       nsmgr);
115             // Get the value of the attribute "ID"
116             string strActivePageID = xmlActiveSection.
                    GetAttribute("ID");
117
118             return strActivePageID;
119         }
120
121         /**************/
122         /* Destructor */
123         /**************/
```

```
124
125         ~AddInSample()
126         {
127             // Disconnect the proxy
128             proxy.Disconnect();
129         }
130
131     }
132 }
```

Anhang B

Inhalt der DVD-ROM

B.1 Diplomarbeit

Pfad: /

Diplomarbeit-Seifried-Aroo.pdf Diplomarbeit mit „Hyperref“
Diplomarbeit-Seifried-Aroo-print.pdf Diplomarbeit ohne „Hyperref“

B.2 Quellcode

Pfad: /sourcecode

dependencies/ . . . . . benötigte Bibliotheken
DiamondTouchAdapter/ Quellcode des DiamondTouch-Adapters für JazzUp
OneNoteWrapper/ . . . Quellcode des OneNote-Wrappers
TangibleFolder/ . . . . Quellcode der Tischanwendung und des Add-ins

B.3 Ausführbare Dateien

Pfad: /binary

Aroo-Add-in/ . . . . . Installer für das OneNote Add-in
Aroo-Table/ . . . . . Tischanwendung
Readme.txt . . . . . kurze Verwendungserklärung

B.4 Sonstiges

Pfad: /

videos/ . . . . . Videos

abbildungen/ . . . . . Bilder und Grafiken
online-quellen/ . . . . . Kopie von Online-Literatur

Literaturverzeichnis

[1] Apple Computer Inc.: *Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley, Reading, MA, USA, 1987.

[2] Arregui, D., C. Fernstrom, F. Pacull, G. Rondeau, J. Willamowski, E. Crochon und F. Favre-Reguillon: *Paper-based Communicating Objects in the Future Office*. Proc. Smart Object Conference, 2003.

[3] Bang, M., A. Larsson und H. Eriksson: *NOSTOS: A Paper-Based Ubiquitous Computing Healthcare Environment to Support Data Capture and Collaboration*. AMIA Annual Symposium Proc., S. 46–50, 2003.

[4] Candib, S.: *What's New for Developers in OneNote 2007*. msdn2.microsoft.com/en-us/office/ms406042.aspx, Mai 2006. Kopie auf DVD-ROM (online-quellen/Candib-MSDN1/OneNote2007-Part1.pdf und OneNote2007-Part2.pdf).

[5] Clark, J., S. DeRose und others: *XML Path Language (XPath) Version 1.0*. W3C Recommendation, 16:1999, 1999.

[6] Cutler, L. D., B. Fröhlich und P. Hanrahan: *Two-handed direct manipulation on the responsive workbench*. In: *SI3D '97: Proc. of the 1997 symposium on Interactive 3D graphics*, S. 107–114, New York, NY, USA, 1997. ACM Press.

[7] Dietz, P. und D. Leigh: *DiamondTouch: a multi-user touch technology*. Proc. of the 14th annual ACM symposium on User interface software and technology, S. 219–226, 2001.

[8] Ecma International: *Standard ECMA-372 - C++/CLI Language Specification*. Techn. Ber., Dezember 2005.

[9] Ecma International: *Standard ECMA-335 - Common Language Infrastructure (CLI)*. Techn. Ber., Juni 2006.

[10] Elwart-Keys, M., D. Halonen, M. Horton, R. Kass und P. Scott: *User interface requirements for face to face groupware*. In: *CHI '90: Proc. of the SIGCHI conference on Human factors in computing systems*, S. 295–301, New York, NY, USA, 1990. ACM Press.

[11] Escapa, D.: *How to create a OneNote 2007 Toolbar Addin*. blogs.msdn.com/descapa/archive/2006/08/31/734298.aspx, 2006. Kopie auf DVD-ROM (online-quellen/Escapa-OneNote/OneNote-Toolbar-Addin.pdf).

[12] Esenther, A., C. Forlines, K. Ryall und S. Shipman: *DiamondTouch SDK: Support for Multi-User, Multi-Touch Applications*. ACM Conference on Computer Supported Cooperative Work (CSCW 2002 Demonstration), 48, 2002.

[13] Fiala, M.: *ARTag, a Fiducial Marker System Using Digital Techniques*. In: *CVPR '05: Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, S. 590–596, Washington, DC, USA, 2005. IEEE Computer Society.

[14] Gamma, E., R. Helm, R. Johnson und J. Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995.

[15] Guimbretière, F.: *Paper augmented digital documents*. In: *UIST '03: Proc. of the 16th annual ACM symposium on User interface software and technology*, S. 51–60, New York, NY, USA, 2003. ACM Press.

[16] Ishii, H. und B. Ullmer: *Tangible bits: towards seamless interfaces between people, bits and atoms*. Proc. of the SIGCHI conference on Human factors in computing systems, S. 234–241, 1997.

[17] Johnson, W., H. Jellinek, J. Leigh Klotz, R. Rao und S. Card: *Bridging the paper and electronic worlds: the paper user interface*. In: *INTERCHI '93: Proc. of the INTERCHI '93 conference on Human factors in computing systems*, S. 507–512, Amsterdam, The Netherlands, 1993. IOS Press.

[18] Kato, H. und M. Billinghurst: *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. Proc. of the 2nd IEEE and ACM International Workshop on Augmented Reality, 99:85–94, 1999.

[19] Kobayashi, M. und H. Koike: *Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system*. ACM Trans. Computer-Human Interaction, 8(4):307–322, 2001.

[20] Koike, H., Y. Sato, Y. Kobayashi, H. Tobita und M. Kobayashi: *Interactive textbook and interactive Venn diagram: natural and intuitive interfaces on augmented desk system*. In: *CHI '00: Proc. of the SIGCHI conference on Human factors in computing systems*, S. 121–128, New York, NY, USA, 2000. ACM Press.

[21] Kruger, R., S. Carpendale, S. D. Scott und A. Tang: *Fluid integration of rotation and translation*. In: *CHI '05: Proc. of the SIGCHI conference on Human factors in computing systems*, S. 601–610, New York, NY, USA, 2005. ACM Press.

[22] Leach, P., M. Mealling und R. Salz: *A Universally Unique IDentifier (UUID) URN Namespace*. 2005.

[23] Leigh, D. und P. Dietz: *DiamondTouch Characteristics and Capabilities*. Proc. of the UbiComp 2002 Workshop on Collaboration with Interactive Walls and Tables, 2002.

[24] Liao, C., F. Guimbretière und K. Hinckley: *PapierCraft: a command system for interactive paper*. In: *UIST '05: Proc. of the 18th annual ACM symposium on User interface software and technology*, S. 241–244, New York, NY, USA, 2005. ACM Press.

[25] Malone, T. W.: *How do people organize their desks?: Implications for the design of office information systems*. ACM Trans. Information Systems, 1(1):99–112, 1983.

[26] Maxell: *Digital Pens: Products information*. www.maxell.co.jp/e/products/industrial/digitalpen/products.html, Juni 2007. Kopie auf DVD-ROM (online-quellen/Maxell-DigitalPens/DigitalPen.pdf).

[27] Maxell: *Digital Pens: Products information: FAQ*. www.maxell.co.jp/e/products/industrial/digitalpen/faqs.html, Juni 2007. Kopie auf DVD-ROM (online-quellen/Maxell-FAQ/DigitalPen-FAQ.pdf).

[28] Microsoft: *List of Bluetooth radio drivers that are included in Windows XP SP2*. support.microsoft.com/kb/841803, Juni 2007. Kopie auf DVD-ROM (online-quellen/Microsoft-Bluetooth/Bluetooth-Driver.pdf).

[29] Mitchell, G.: *Orientation on Tabletop Displays*. Diplomarbeit, Simon Fraser University, 2003.

[30] Philipose, M., K. Fishkin, D. Fox, D. Hahnel und W. Burgard: *Mapping and Localization with RFID Technology*. In: *ICRA '04: Proc. of the IEEE International Conference on Robotics and Automation, 2004*, May 2004.

[31] Rao, R., S. K. Card, W. Johnson, L. Klotz und R. H. Trigg: *Protofoil: storing and finding the information worker's paper documents in an electronic file cabinet*. In: *CHI '94: Proc. of the SIGCHI conference on Human factors in computing systems*, S. 180–185, New York, NY, USA, 1994. ACM Press.

[32] Rekimoto, J. und M. Saitoh: *Augmented surfaces: a spatially continuous work space for hybrid computing environments*. In: *CHI '99: Proc. of the SIGCHI conference on Human factors in computing systems*, S. 378–385, New York, NY, USA, 1999. ACM Press.

[33] Scott, S., K. Grant und R. Mandryk: *System Guidelines for Co-located, Collaborative Work on a Tabletop Display*. Proc. ECSCW 2003, S. 159–178.

[34] Sellen, A. J. und R. H. Harper: *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.

[35] Shneiderman, B.: *Designing the user interface (2nd ed.): strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[36] Ullmer, B., H. Ishii und D. Glas: *mediaBlocks: physical containers, transports, and controls for online media*. In: *SIGGRAPH '98: Proc. of the 25th annual conference on Computer graphics and interactive techniques*, S. 379–386, New York, NY, USA, 1998. ACM Press.

[37] United States National Museum: *Report on the Progress and Condition of the U.S. Nation Museum for the Year Ending June 30*, S. 66. U.S. Government Printing Office, 1939.

[38] Van Laerhoven, K., A. Schmidt und H. Gellersen: *Pin&Play: Networking Objects through Pins*. Proc. of Ubicomp, 2498:219–229, 2002.

[39] Van Laerhoven, K., N. Villar, A. Schmidt, H. Gellersen, M. Hakansson und L. Holmquist: *Pin&Play: the surface as network medium*. Communications Magazine, IEEE, 41(4):90–95, 2003.

[40] Villar, N. und H. Gellersen: *A malleable control structure for soft-wired user interfaces*. In: *TEI '07: Proc. of the 1st international conference on Tangible and embedded interaction*, S. 49–56, New York, NY, USA, 2007. ACM Press.

[41] Vogt, H.: *Multiple object identification with passive RFID tags*. 2002 IEEE International Conference on Systems, Man and Cybernetics, 3, 2002.

[42] Wellner, P.: *Interacting with paper on the DigitalDesk*. Commun. ACM, 36(7):87–96, 1993.

[43] Wikipedia: *Unterstützte Sprachen und Compiler (.NET)*. de.wikipedia.org/wiki/Unterstuetzte_Sprachen_und_Compiler_(.NET), Juni 2007. Kopie auf DVD-ROM (online-quellen/Wikipedia-DotNet/DotNetSprachen.pdf).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —

Breite = 100 mm
Höhe = 50 mm

— Diese Seite nach dem Druck entfernen! —