

Diplomarbeit

Sven Wille
Marc Lembcke

Easy Presenter:
eine bluetoothbasierte Multimedia-Anwendung

Studiengang Technische Informatik
Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Gunter Klemke
Abgegeben am 10. Juli 2003

Danksagung

Hiermit möchten wir uns bei allen Leuten bedanken, die uns bei der Diplomarbeit sowohl fachlich als auch seelisch unterstützt haben.

Das sind: Kai, Gunter, Martin, Ina, Thomas, Arne, Ole, Mark, Mellie, Aida, Hartmut, Mirco, Oumar, Papi, Kiet, Alex und ganz besonders bei Mama und Papa!

Und sollten wir noch jemanden vergessen haben: „Auch Dir herzlichen Dank!!!“

Easy Presenter: eine bluetoothbasierte Multimedia-Anwendung

Stichworte Bluetooth, Ubiquitous Computing, Verteilte Systeme, PDA, Multimediastreaming

Zusammenfassung

Diese Diplomarbeit beschäftigt sich mit der Funktechnologie „Bluetooth“. In diesem speziellen Fall geht es um eine Multimedia-Anwendung mit Bluetooth und einem PDA. Diese Anwendung soll es ermöglichen, eine Präsentation auf einem PDA zu speichern, diese zum Vortragsort zu transportieren und dort auf einem Beamer anzuzeigen und fernzusteuern. Der Schwerpunkt liegt dabei im Design dieser Software und der Umsetzung des Multimediastreamings (Übertragung) und Steuerung der Präsentation über die Bluetooth-Verbindung.

Easy Presenter: A Bluetooth based multimedia application

Keywords Bluetooth, Ubiquitous Computing, Distributed Systems, PDA, Multimedia-streaming

Abstract

This thesis deals with the communication technology-“Bluetooth“. This case handles especially an application of Multimedia with Bluetooth and a PDA. The application should make it possible to store a presentation on a PDA; to deliver it to the conference site, and display on a beamer, where is could be controlled from remote. The main focus, thus, lies on the software designing, streaming the application of multimedia and controlling the presentation on the Bluetooth connection.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Motivation und Zielsetzung	5
1.2	Beschreibung der Aufgabe	6
1.3	Einteilung unserer Diplomarbeit	7
1.4	Eingetragene Warenzeichen	7
2	Grundlagen	8
2.1	Ubiquitous Computing	8
2.2	Verteilte Systeme	9
2.3	Bluetooth	15
3	Analyse	29
3.1	PES - Präsentations-Erstellungssystem	30
3.2	PSE - Präsentations-Steuereinheit	31
3.3	PAE - Präsentations-Anzeigeeinheit	38
4	Das Labor	40
4.1	Desktop PC	41
4.2	PDA	43
4.3	Präsentationsmaschine	45
4.4	Erstellungsprogramme	46

	4
5 Design und Realisierung	49
5.1 Systemarchitektur	49
5.2 PES - Präsentations-Erstellungssystem	51
5.3 PSE - Präsentations-Steuerinheit	56
5.4 PAE - Präsentations-Anzeigeeinheit	74
5.5 Zusammenfassung	79
6 Resümee	81
6.1 Erreichte Ziele	81
6.2 Ausblick	82
A Beschreibung der Methoden und Variablen	85
B FAQ	95
C Installationshinweise	97
D Glossar	98
E Evolution	110
E.1 TCP/IP Verbindung	110
E.2 ActiveX Browser in Visual C++ einbinden	113

Kapitel 1

Einleitung

Die Ära des „Ubiquitous Computing“¹ ist angebrochen. Die Anzahl von mobilen Geräten und Sensoren, sowohl im Geschäfts- als auch Privatenbereich, vermehren sich stetig und jeder will so viele Dienste, wie möglich damit nutzen. Damit diese Dienste überhaupt nutzbar werden, müssen diese Geräte und Sensoren miteinander kommunizieren. Hierfür werden Kommunikationstechnologien gesucht, die kabellos, energiesparsam und ohne Sichtkontakt funktionieren.

1.1 Motivation und Zielsetzung

Bei mobilen Geräten spielt die Kommunikation untereinander eine sehr wichtiger Rolle. Da heutzutage aber kein Mensch mehr zusätzlich zu seinem „handheld“²-Gerät ein Kabel bei sich haben will, ist an dieser Stelle eine kabellose Verbindung erforderlich. Hier haben wir uns dazu entschieden, ein Projekt mit der Funktechnologie Bluetooth zu erstellen, da diese alle wichtigen Anforderungen, die oben in der Einleitung erwähnt wurden, erfüllt. Ausserdem hat es sich mittlerweile sehr verbreitet und ist in vielen mobilen Geräten bereits integriert, bzw. leicht nachrüstbar. Daher ist die Entscheidung bei der Suche nach einem „handheld“-Gerät auch sehr schnell auf einen PDA³ mit integriertem Bluetooth-Modul gefallen (in unserem Fall der Compaq iPAQ H3870). Bei der Überlegung, was man mit dem PDA machen könnte und da das Projekt auch einen Nutzen für die HAW-Hamburg haben sollte, ist die Idee für dieses Projekt entstanden:

¹Die deutsche Übersetzung von *Ubiquitous Computing* lautet *allgegenwärtige Computer*, für eine näherer Erklärung siehe Kapitel 2.1

²Tragbare Rechner, die so klein sind, dass sie in einer Hand gehalten werden können, während die andere Hand Daten eingibt

³PDA - Personal Digital Assistant

Ein Gerät, welches man immer bei sich hat (PDA), mit Hilfe einer drahtlosen Übertragungstechnologie (Bluetooth), als Fernbedienung und Transportmittel für eine Präsentation zu benutzen.

Die Handhabung des „Easy Presenter“⁴ soll hierbei möglichst einfach sein. Das bedeutet, es soll möglich sein, nach Betreten des Vortragsraums sofort mit der Präsentation beginnen zu können.

1.2 Beschreibung der Aufgabe

Die Aufgabenstellung lässt sich durch folgendes Szenario am besten beschreiben: Es soll möglich sein, eine Präsentation oder Diashow zu erstellen und diese als einzelne Bilder abzuspeichern. Danach sollen diese Bilddateien auf eine mobiles Gerät übertragen und in einen vordefinierten Ordner abgespeichert werden. Auf dem Weg zu Vortragsort soll der Anwender sich die Präsentation auf seinem mobilen Gerät anschauen, sowie eine Diashow aus den zuvor abgespeicherten Bilddateien erstellen können. Ausserdem soll er die Möglichkeit haben, eine erstellte Diashow als Profil abzuspeichern und zu laden. Nachdem der Anwender am Vortragsort angekommen ist, soll er ohne weiteren Aufwand die Präsentations-Dateien über eine Funkverbindung auf ein Anzeigegerät übertragen und dann die Präsentation oder Diashow auf einen Beamer anzeigen lassen können. Dabei soll die Präsentation über diese kabellose Verbindung, mit Hilfe des Gerätes, ferngesteuert werden. Die Abbildung 1.1 gibt einen ersten Überblick über die Aufgabenstellung und den Ablauf.

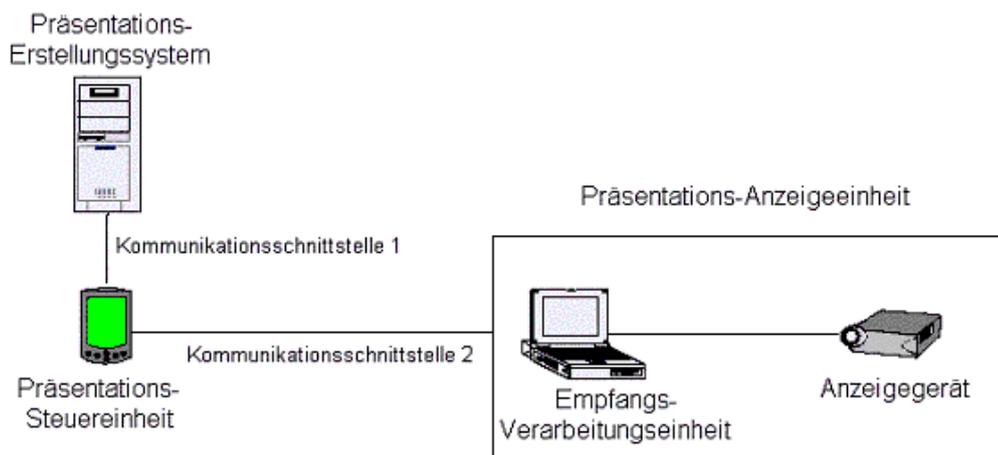


Abbildung 1.1: Gesamtüberblick der Systemarchitektur - 1. Version

⁴Easy Presenter - Zu entwickelnde Software für die Präsentationsfernbedienung.

1.3 Einteilung unserer Diplomarbeit

Da die Diplomarbeit von Herrn Sven Wille und Herrn Marc Lembcke gemeinsam geschrieben wurde, wird hier klar gekennzeichnet, wer für welche Kapitel die Verantwortung hat:

Sven Wille: Kapitel 2.1, 2.2, 3.2, 4.2, 5.1, 5.2, 5.3.1(Sequenzdiagramme), 6, B, C, D, E

Marc Lembcke: Kapitel 1, 2.3, 3.1, 3.3, 4.1, 4.3, 4.4, 5.3.1(Klassendiagramm, Systemkomponenten), 5.3.2, 5.4, 5.5, A

1.4 Eingetragene Warenzeichen

BluetoothTM ist ein eingetragenes Warenzeichen der Bluetooth SIG Inc.

Compaq, iPAQ und das iPAQ Pocket-PC-Produktdesign sind Warenzeichen der Hewlett-Packard Corporation.

TDK ist ein eingetragenes Warenzeichen der © Copyright 2003 TDK Electronics Europe GmbH.

Windows[®], Visual C++[®], Powerpoint[®], WinCE[®] und PPC 2002[®] sind eingetragene Warenzeichen der Microsoft[®] Inc.

Together[®] 6.0 ist ein eingetragenes Warenzeichen der Copyright ©2002 TogetherSoft Corporation.

BTAcess for the iPAQTM ist ein eingetragenes Warenzeichen der Copyright ©2002-03 High Point Software.

Kapitel 2

Grundlagen

2.1 Ubiquitous Computing

2.1.1 Begriff

Der Begriff des Ubiquitous Computing wurde vor gut 10 Jahren von Mark Weiser, dem Entwicklungsleiter von Xerox PARC, geprägt.

Weiser geht davon aus, dass zukünftig verschiedenste Alltagsgegenstände mit miniaturisierter, eigener Rechenleistung ausgestattet sein werden, die Sensoren besitzen, um ihre Umgebung zu erfassen. Ausserdem werden sie mit anderen Gegenständen kommunizieren, um Informationen gegenseitig auszutauschen. Aber trotzdem werden die von uns alltäglich genutzten Mini-Rechner, Mikrochips und Sensoren, für uns nicht mehr sichtbar, immer mehr in den Hintergrund geraten. Daraus folgt auch die deutsche Übersetzung „Allgegenwärtige Computer“, die allerdings nicht so häufig genutzt wird. [1]

2.1.2 Historie

Mark Weiser definiert, die 3 Dekaden der Computerentwicklung wie folgt [2]:

„Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives. Alan Kay of Apple calls this „Third Paradigm“ computing.“ [2]

Heute im Jahr 2003, gibt es schon viele fertig entwickelte Anwendungsmöglichkeiten in der Mobilkommunikation, z.B. ein Headset mit Bluetooth fürs Handy.

2.1.3 Hohes Marktpotenzial

Ein Hauptanwendungsbereich für Ubiquitous Computing ist „Mobile Computing“. Aus der Pressemitteilung vom VDE [3] wird ersichtlich, dass gerade in der Mobilindustrie ein großes Marktpotential vorhanden ist:

...„Ubiquitous Computing“, die allgegenwärtige Verfügbarkeit von Mini-Computern, Mikrochips und Sensoren wird in der Informationstechnik und ihren Anwendungen völlig neue Möglichkeiten eröffnen. Nach Einschätzung des VDE haben Deutschland und Europa eine hervorragende Technologie- und Marktposition auf diesem Gebiet. Eine hochmoderne Infrastruktur in der Mobilkommunikation, der Start von UMTS sowie Stärken in den Mikro- und Systemtechniken sind entscheidende Gründe dafür. Der Preis für die Minirechner, mit denen man beispielsweise Kleidung oder Brillen ausstattet, könnte schon bald bei nur 50 Cent liegen.

Eine weitere Voraussetzung ist eine hohe Technikakzeptanz. Nach einer zur CeBIT 2002 vorgestellten repräsentativen VDE- Studie liegt die Zustimmung zur Informationsgesellschaft bei 51% der Bevölkerung, nur 9% sind eher skeptisch. Besonders junge Menschen, 62% der bis 34-jährigen, sind für schnelle Innovationen in der Informationstechnik und Telekommunikation. Das sind 10% mehr als noch vor vier Jahren... [3]

Eine weitere Bestätigung für die Akzeptanz in der Mobilkommunikation zeigen die Zahlen im Werkstattbericht Nr.49 „Entwicklung und zukünftige Bedeutung mobiler Multimedien“ [4]. Auch hier liegt der Bedarf in Deutschland im Jahre 2003 über 50%.

Um einen Eindruck über die ansteigenden Mobilfunkraten in Europa zu bekommen, siehe Abbildung 2.1

2.2 Verteilte Systeme

Warum reden wir über Verteilte Systeme (VS)?

Durch die Tatsache, dass wir einen PDA als Fernbedienung und als Transportmittel für unsere Präsentation benutzen¹, haben wir unser System auf mehrere „Rechensysteme“

¹Weitere Information, Kapitel 1.2

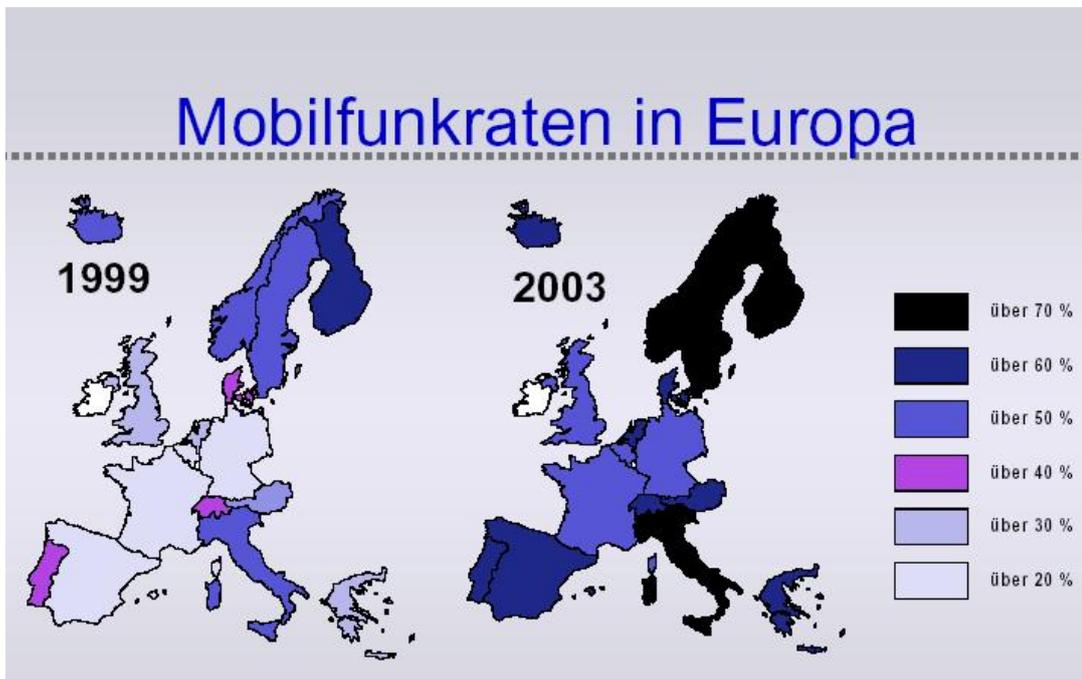


Abbildung 2.1: Vergleich der Mobilfunkraten, von 1999 und 2003, in Europa

verteilt. Deshalb reden wir im Zusammenhang mit unserer Diplomarbeit von „Verteilten Systemen“ in die wir in diesem Kapitel einen kleinen Einblick geben wollen. Für weitere Informationen zu Verteilten Systemen siehe Tannenbaum [5].

2.2.1 Definition

Tannenbaum definiert „Verteilte Systeme“ folgendermaßen:

„A distributed system is a collection of independent computers that appears to its users as a single coherent system.“ [5]

Eine etwas lustigere Definition von Leslie Lamport lautet:

„A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.“ [5]

2.2.2 Middleware

Die Middleware unterstützt die Entwicklung und Ausführung von verteilten Anwendungen. Sie bildet eine Zwischenschicht zwischen den Anwendungen und dem Netzwerk und stellt dem Entwickler Dienste zur Verfügung, die dieser nutzen kann. Dadurch braucht der Entwickler nichts über den Ort oder die Funktionalitäten der aufzurufenden Funktionen zu wissen. Diese Zwischenschicht (Middleware - Plattform) wird daher auch Verteilungsplattform genannt. Beispiele für Middleware sind: CORBA (Common Object Request Broker Architecture) oder .net (Microsoft). Die Middleware CORBA z.B. befreit den Entwickler von den Details der Kommunikation zwischen den Objekten. Es ist völlig transparent, auf welcher Hardware, mit welchem Betriebssystem, auf welchem Knoten im Netz sich die Kommunikationspartner befinden, in welcher Programmiersprache sie implementiert sind oder welche Netzstruktur eingesetzt wird.

2.2.3 Client-Server Modell

Da in dieser Diplomarbeit ein Client-Server Modell verwendet wird, wird an dieser Stelle ein kurzer Einblick in dieses Modell gegeben. Das Client-Server Modell ermöglicht netzfähige Anwendungen in verteilten Systemen mit Leistungsmerkmalen, die nach außen den Anschein eines einzigen geschlossenen Systems widerspiegeln. Es gibt zwei verschiedene Rollen in diesem Modell:

1. Client - ruft einen Dienst auf
2. Server - bietet einen Dienst an

Bevor eine Verbindung zwischen Client und Server aufgebaut werden kann, müssen sie sich zuerst über ein Standardprotokoll einigen, z.B. TCP oder UDP². Ausserdem können auch mehrere Clients auf einen Server zugreifen. In dieser Diplomarbeit z.B. kommuniziert der PDA (Client) mit der Präsentationsmaschine (Server). Desweiteren soll es möglich sein, dass mehrere PDAs Dienste auf der Präsentationsmaschine aufrufen können.

Die Client-Server Anwendungsdienste interagieren hierbei auf Schicht 7 (Anwenderschicht) gem. OSI-Referenzmodell³.

Internetanwendungen werden nach dem Client-Server-Prinzip erstellt. Der Client wird in den meisten Fällen als „Benutzer-Interface“ realisiert und nimmt vom Server bestimmte Dienste in Anspruch. Er baut in Abhängigkeit vorher definierter Ereignisse, z.B. dem Starten einer Internetanwendung durch einen Anwender, die Verbindung zum Server auf

²UDP - User Datagram Protocol - Es handelt sich hierbei um ein verbindungsloses Transportprotokoll

³OSI - Open Systems Interconnection [22]

und ist somit der aktive Teil. Der Server stellt nun den gewünschten Dienst zur Verfügung. Er muß sich ständig in einem Zustand befinden, in dem er Verbindungsaufforderungen von Clients entgegennehmen kann, er ist der passive Teil. Ein Server darf niemals einen Dienst vom Client anfordern. Client und Server müssen die gleiche Sprache sprechen: Sie müssen sich also an einem gemeinsamen Protokoll orientieren.

Das unterschiedliche Verhalten von Client und Server läßt allerdings eine Asymmetrie entstehen, die sich in der Verwendung unterschiedlicher Schnittstellen-Befehle bei der Realisierung einer Client- oder Serverapplikation niederschlägt.

2.2.4 Kommunikationsfähigkeit

Was braucht man, um autonome Rechnersysteme in einem VS kommunizieren zu lassen?

1. Man braucht ein physikalisches Übertragungsmedium, über den die Rechnersysteme in Kontakt treten können
2. Es wird sich über ein Standard (Kontrollprotokoll) geeinigt
3. Es müssen Verabredungen über das ausgetauschte Format (Syntax) getroffen werden

Zu 1.: Was für physikalische Übertragungsmedien gibt es?

Auf der physikalischen Schicht im ISO-OSI, werden die Informationen auf der untersten Schicht bitorientiert über ein beliebiges Medium übertragen.

Es gibt folgende Übertragungsmöglichkeiten: Für den Einsatz von mobilen Geräten, z.B.:

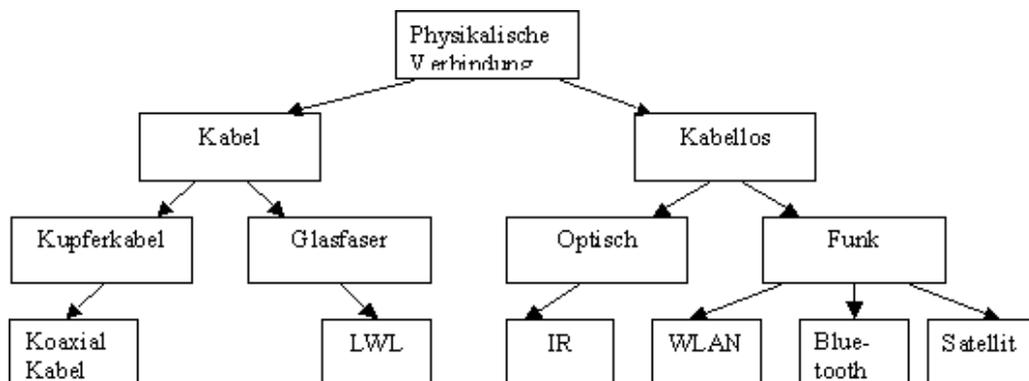


Abbildung 2.2: Physikalische Medien

PDA, Handy, usw., ist eine kabellose Verbindung besser geeignet, da man sich hiermit unabhängig vom Kabel in einen Raum bewegen kann. Bei den kabellosen Verbindungen stehen zur Zeit zwei physikalische Technologien zur Verfügung: Optisch (z.B.: Infrarot) und Funk (z.B.: WLAN, Bluetooth)

Bei der Optischen Übertragung von Daten brauchen die Geräte, die miteinander in Verbindung treten wollen direkten Sichtkontakt. Dies ist ein wesentlicher Nachteil für mobile Geräte, da nicht immer der direkte Sichtkontakt gewährleistet werden kann. Die beiden folgenden Beispiele sollen diese These untermauern:

1. Eine Person X will mit einem PDA über das Internet eMail's abrufen. Er muss hierfür eine Verbindung vom PDA zum Handy aufbauen, um sich über das Handy ins Internet einwählen zu können. Das Handy befindet sich aber im Aktenkoffer oder in der Jackentasche, also besteht kein direkter Sichtkontakt vom PDA zum Handy, um eine Verbindung über Infrarot aufbauen zu können.
2. Etwas spezieller: In unserer Diplomarbeit wollen wir eine Präsentation vom PDA zu einem Anzeigegerät übermitteln und anzeigen lassen. Auch hier ist nicht immer ein direkter Sichtkontakt vorhanden, da sich die vortragende Person mit dem PDA im Raum bewegt und kein direkter Sichtkontakt die ganze Zeit zwischen PDA und Anzeigegerät gewährleistet werden kann.

Die Konsequenz ist, dass wir eine Technologie benutzen wollen, die kein Kabel und kein direkten Sichtkontakt benötigt. Die Lösung ist die Verwendung der Funktechnologie zur Übertragung der Daten. Zur Zeit gibt es, für unsere Diplomarbeit, zwei geeignete Funktechnologie-Standards auf dem Markt: WLAN (Wireless Local Area Network) und Bluetooth (weitere Daten über diese zwei Standards, Tabelle 2.1 Der Energieverbrauch in unserer Diplomarbeit spielt eine wesentliche Rolle, da die Stromversorgung des PDA über Batterie läuft und eine Präsentation auch über eine halbe Stunde gehen sollte, ohne dass die Batterie leer wird. Dadurch kommen wir zur Erkenntnis, dass Bluetooth durch seinen niedrigen Energieverbrauch (ca.1 mW) zu WLAN (ca.100 mW) am Besten geeignet ist.

Zu 2.: Desweiteren müssen sich beide Seiten über ein Kontrollprotokoll, wie z.B. TCP/IP, UDP, usw. einigen.

Zu 3.: Nachdem geklärt ist, über welches Medium mit der eingesetzten Technologie (Bluetooth) die Übertragung zwischen den verteilten Rechnersystemen ablaufen soll, und sich über ein Kontrollprotokoll geeinigt wurde, müssen sich beide Seiten noch über ein Format (z.B. XML, HTML, JPG, usw.) und deren Semantik einigen.

	WLAN	Bluetooth
Technologie	IEEE-Standard 802.11b	IEEE 802.15
Frequenzband	2,4 GHz	2,4 GHz
Transferrate	11 Mbit/s	723 kbit/s
Übertragungsverfahren	Direct Sequence Spread Spectrum (DSSS)	Frequenzsprungverfahren-Frequency Hopping Spread Spektrum (FHSS)
Schutzmechanismen	Wired Equivalent Privacy (WEP)	Frequenzwechsel 128 Bit verschlüsselt
Reichweite	10-300m	10m (100m Long Range BT)
Einsatzgebiet	Temporäre Vernetzung: z.B. auf Messeständen oder zur einfachen Integration von Notebooks in Firmennetzwerke	Vernetzung von PDA, Handys oder anderen Devices AddHoc Netzwerkaufbau (keine bekannte IP-Adresse nötig!)
Energiebedarf	100 mW	1 mW

Tabelle 2.1: Vergleich WLAN mit Bluetooth

2.2.5 Konsequenzen für unsere Arbeit

Ein entscheidender Vorteil von Verteilten Systemen ist, dass das geplante Projekt ohne Probleme in Zukunft durch neue Features erweitert werden kann. Wie z.B. mit dem im Ausblick beschriebenen Szenario, dass sich mehrere Zuhörer in das bestehende Netzwerk einbinden können, um vielleicht die vorgeführte Präsentation von dem Server auf ihrem PDA herunterzuladen oder um vielleicht Kommentare in Form von Notizen auf der an die Wand projizierten Folie machen zu können. Desweiteren kann später der PDA als Fernbedienung durch ein anderes Gerät mit ähnlicher Technik (z.B. Web Pad oder Web Tablet) ohne Probleme ersetzt werden.

Verteilte Systeme sind nützlich um gemeinsame Ressourcen verwenden zu können. Die Ressourcen sind in die Bereiche Hardware und Software aufgeteilt.

Eine sinnvolle Hardwareressource ist z.B. ein Drucker, der von verschiedenen, räumlich getrennten, Arbeitsplätzen angesprochen werden kann. In unserer Diplomarbeit wäre die Präsentationsmaschine eine sinnvolle Hardwareressource, da jeder Client (z.B. ein PDA) seine Präsentationsfolien über diese Ressource anzeigen lassen könnte.

Ein Beispiel für gemeinsam genutzte Softwareressourcen ist das Internet, wo Webinhalte gleichzeitig von mehreren Benutzern gelesen werden können.

Auch in unserer Diplomarbeit wäre die Nutzung dieser Ressource denkbar, z.B. könnten mehrere Clients, Zuhörer mit Ihren PDA's, sich die angezeigten Präsentationsfolien auf den eigenen PDA herunterladen.

2.3 Bluetooth



Abbildung 2.3: Größenvergleich: Bluetooth Chip mit 50 Pfennig Stück [14]
Links ein aktuelles HF-Modul von National Semiconductor (LMX3162), rechts ein künftiges Komplett-Bluetooth-Modul.

2.3.1 Historie

Der Name „Bluetooth“, oder zu deutsch Blauzahn, geht zurück auf den Wikinger König Harald Blåtand, auch Blauzahn genannt, der im zehnten Jahrhundert durch seine - für damalige Verhältnisse - außergewöhnliche Fähigkeit zu kommunizieren, Dänemark und Norwegen vereinte, dort den christlichen Glauben einführte und eine große Vorliebe für Blaubeeren hatte, die zu einer entsprechenden Verfärbung seiner Zähne führte. Als Symbol für Verständigung, Erfolg und Sicherheit, und wegen des historischen Bezugs zu Skandinavien wählte Ericsson schließlich Bluetooth als Codenamen.

Bluetooth geht zurück auf einen Entwurf, der bereits 1994 von Ericsson entwickelt wurde, um die mechanisch anfälligen und wenig komfortablen Kabelverbindungen zwischen Kommunikationsprodukten zu ersetzen. Um von Umgebungs- und Betriebsbedingungen möglichst unabhängig zu sein, wurde die Funktechnologie gegenüber der zu diesem Zeitpunkt bereits populäreren Infrarotübertragung favorisiert. Damit wurden Verbindungen auch durch Stoff, Leder, Wände und ohne direkten Sichtkontakt möglich.

Ericsson trat nach der grundsätzlich Konzeption der Technologie an andere Hersteller von tragbaren elektronischen Geräten heran, um einen weltweiten Standard zu entwickeln. 1998 wurde gemeinsam mit Nokia, IBM, Toshiba und Intel die Bluetooth Special Interest Group (SIG) gegründet, die schon im Mai des selben Jahres erstmals an die Öffentlichkeit trat und der sich mittlerweile mehr als 2100 Unternehmen durch eine Bluetooth-Anwender-Übereinkunft angeschlossen haben.

2.3.2 Allgemeine Grundlagen

Die allgemeinen Grundlagen sind dem „Bluetooth Whitepaper“ [11] entnommen.

Jedes Bluetoothfähige Gerät kann innerhalb eines drahtlosen Kleinnetzes, eines sogenannten Piconet, mit einer Reihe von anderen Geräten kommunizieren. Alle Geräte unterhalten eine logische Verbindung untereinander, die nur im Falle der tatsächlichen Datenübertragung in eine physikalische Verbindung verwandelt wird.

Mehr als zwei Bluetooth Systeme bilden ein Piconet, in dem eine Einheit als Master und die anderen als Slave arbeiten. Der Master ist immer die Einheit die zuerst eine Verbindung initiiert (PAGE device). Bis zu sieben Slaves können in einem Piconet aktiv sein. Zusätzlich können bis zu 255 Slaves im sogenannten Park-modus im Piconet eingebucht sein. Diese parkenden Bluetooth Komponenten können von sich aus keine Daten senden, synchronisieren sich aber laufend mit dem Master und können von diesem aus dem Parkmodus in den regulären Kommunikationsstatus aktiviert werden.

Viele Bluetooth Anwendungen, die als Profile bezeichnet werden, werden aus technischen Gründen den Piconet Modus nicht unterstützen: Bei der Fax-Übertragung z.B. müssen in der Kommunikation enge Zeitfenster eingehalten werden, die beim gleichzeitigen Betrieb einer Piconet Applikation nicht garantiert werden können. Daher ist die Fax Übertragung eine ausschließliche Point-to-Point Verbindung.

Mehrere Piconets mit sich überlappenden Bereichen werden als Scatternet bezeichnet. In jedem Piconet kann nur ein Master festgelegt werden, aber Slaves können in mehreren Netzen eingebucht sein (Abb. 2.4). Die notwendigen Spezifikation für die Verwaltung eines Scatternets sind sehr umfangreich und zum jetzigen Zeitpunkt noch nicht in vollem Umfang festgelegt.

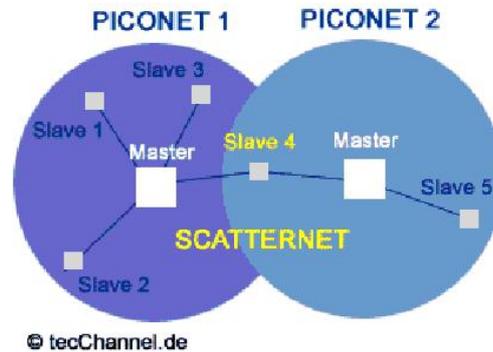


Abbildung 2.4: Scatternet mit 2 Piconets im Verbund [14]

Bluetooth arbeitet in einer Frequenz zwischen 2,402 GHz und 2,480 GHz, dem ISM-Band (Industrial, Scientific, Medical), das mittlerweile weltweit frei und unlicenziert ist. Dieses ISM-Band wird für das Frequenz Hopping Verfahren in 79 Kanäle zu je 1 MHz Bandbreite aufgeteilt.

Bei der Übertragung wird das Fast Frequency Hopping Verfahren eingesetzt, bei dem 1600 Frequenzwechsel pro Sekunde vollzogen werden, um Abhörsicherheit und eine große Störfestigkeit gegenüber anderen Einstrahlungen in diesem Frequenzbereich zu gewährleisten.

Die Identifizierung der einzelnen Bluetooth-Komponenten erfolgt durch eine eindeutige Adresse, ähnlich der MAC-Adresse⁴ im Ethernet Netzwerk. Dadurch werden Rechte, Funktionen und Sicherheit der miteinander kommunizierenden Teilnehmer geregelt.

Das Bluetooth Protokoll unterstützt einen asynchronen Datenkanal, bis zu drei synchronen Sprachkanälen oder einen Kanal, über den gleichzeitig Daten und Sprache übertragen werden können. Jeder Sprachkanal hat eine Bandbreite von 64 kB/s in beiden Richtungen. Der Datenkanal hat eine maximale Kapazität von 723,3 kB/s, wobei gleichzeitig noch bis zu 57,6 kB/s für einen Rückkanal zur Verfügung stehen. Dieser Modus eignet sich beispielsweise besonders für Internet-Anwendungen, weil dabei die Daten mit gut 700 kB/s aus dem Internet geladen werden und gleichzeitig noch genug Kapazität für Eingaben vorhanden ist. Die Gesamtkapazität kann aber auch z.B. symmetrisch auf zwei Kanäle zu je 172,8 kB/s verteilt werden.

Jedes Bluetooth Device (Gerät) erhält eine eindeutige Adresse von 48 Bit Länge, deren Aufbau nach dem IEEE 802 Standard geregelt ist. Die Adresse ist hardwaremäßig im ROM des Bluetooth Moduls festgelegt und kann nicht verändert werden.

⁴Hardware-Adresse eines jeden LAN-Controllers, die zur eindeutigen Identifikation eines Knotens im Netzwerk dient. Die MAC-Adresse wird fest in einem Chip eingebrannt und kann nicht mehr verändert werden.

Sobald Bluetooth aktiviert wird, sendet es Suchsignale (Requests) aus, um andere Teilnehmer zu finden. Wenn ein Request beantwortet ist, tauschen die beiden Bluetooth Systeme ihre MAC-Adress-Informationen aus. Zu Beginn der Kommunikation zwischen den beiden Komponenten muß der Anwender meistens durch Eingabe einer PIN (Personal Identification Number) die Verbindung freigeben. Diese PIN kann wahlweise einmalig für diese Kommunikation oder bei jedem Verbindungsaufbau angefordert werden.

Danach tauschen die Bluetooth Module sogenannte Profile aus, in denen festgelegt ist, welche Funktionen die beiden Endgeräte sich gegenseitig über die Funkverbindung zur Verfügung stellen und welche Daten oder Parameter dafür erforderlich sind.

Die Reichweite einer Bluetooth-Verbindung liegt bei maximal 10 Meter (1mW;0db), es befindet sich aber bereits eine sogenannte 20 dB (100mW) Spezifikation in Vorbereitung, die bis zu 100 Meter Übertragung leisten wird, dann aber einen entsprechend höheren Stromverbrauch hat. Ein Vergleich mit Mobiltelefonen zeigt, wie gering die Sendeleistung von Bluetooth-Geräten ist: Die kleinsten Mobiltelefone haben eine Sendeleistung von 0,8 Watt, also um den Faktor 800 höher als ein Bluetooth-Gerät mit 1mW.

Weitere Vorteile von Bluetooth sind:

Größere Unempfindlichkeit gegenüber Störstrahlung:

Ist ein Frequenzband durch eine Störstrahlung (z.B. Mikrowellenherde "senden" in diesem Bereich) blockiert, ist nur für eine 1/1600 Sekunde die Übertragung unterbrochen, die Performance der Funkverbindung beträgt dann immer noch 1599/1600 bzw. 99,93%, wenn tatsächlich nur ein Kanal betroffen ist.

Höhere Sicherheit:

Die schnellen Frequenz-Wechsel (oder auch "Frequenz-Hoppin" genannt) und der dahinter liegende Algorithmus sind nur mit sehr aufwendiger Technik zu erfassen und kaum zu decodieren.

Dominant gegenüber anderen Funkverbindungen:

Bluetooth kann sich gegenüber anderen Funkverbindungen durchsetzen: Trifft eine DECT⁵-Modul auf ein blockiertes Frequenzband, wartet es eine Sekunde, bis der Sprung auf das nächste Band erfolgt. Da Bluetooth 1600 Mal pro Sekunde springt, ist auch die Wahrscheinlichkeit, daß Bluetooth eine DECT-Übertragung blockiert 1600 Mal höher, als daß DECT Bluetooth stört.

Wegen des geringen Schutzes des Funknetzes gegenüber herkömmlichen Kabelverbindungen hat die Verschlüsselung und Datensicherheit im Bluetooth Standard einen besonders hohen Stellenwert. Aus der Bluetooth Adresse und einem Zufallsgenerator wird ein Schlüssel bestimmt, der über den Master die Hopping Sequenz festlegt, die alle weiteren Piconet Teilnehmer nutzen müssen. Wegen des Zufallsgenerators gilt auch für häufig miteinander kommunizierende Bluetooth Komponenten für jede neue Verbindung ein neuer

⁵DECT (Digital European Cordless Telecommunications) ist ein weiterer Standard für drahtlose Funktechnologie (weitere Informationen siehe: <http://www.tecchannel.de/hardware/511/>)

Hopping Algorithmus. Die über die ständig wechselnden Frequenzbänder übertragenen Daten werden zusätzlich mit einem 128 Bit Schlüssel kodiert. Das Ziel der Entwickler ist, die Bauform des Baseband und des Bluetooth Radio mit möglichst geringen Abmessungen (elektrische Komponenten / Chips) zu realisieren, damit Bluetooth auch in kleinsten Geräten (Handys, Schlüsselanhänger) integriert werden kann.

2.3.3 Verbindungsaufbau zwischen Bluetooth Geräten

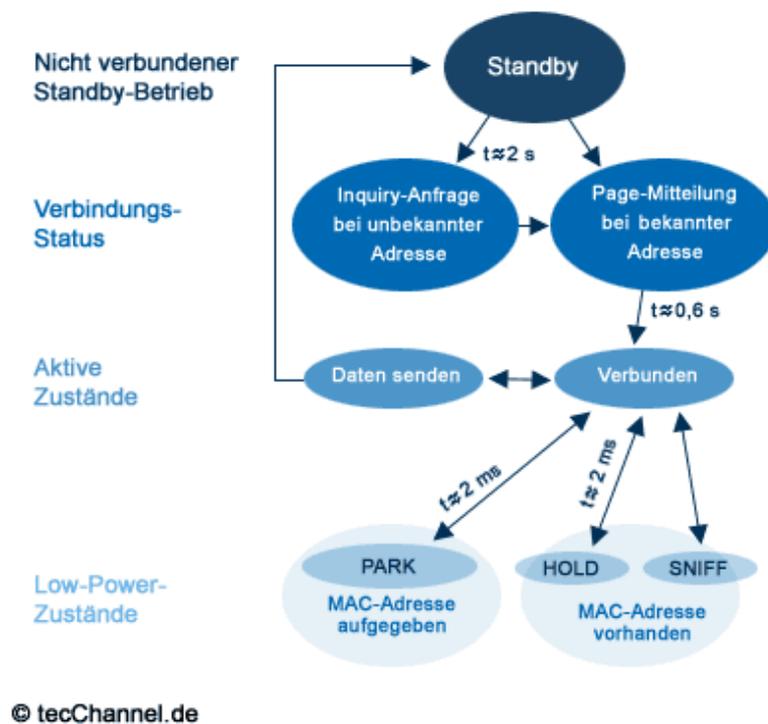


Abbildung 2.5: Verbindungsaufbau Bluetooth [14]

Das Bild zeigt die Übergänge zwischen den einzelnen Zuständen: Standby, Inquiry, Page, Senden, Verbunden, Park, Hold und Sniff

Die Kommunikation läuft wie folgt ab (Abbildung 2.5):

1. Wenn die Geräte eingeschaltet sind, befinden diese sich im Standby Mode. Hier wird alle 1,28 Sekunden überprüft, ob Nachrichten anliegen. Dabei werden 32 Frequenzsprünge auf Daten für diese Station überprüft. Stationen in diesem State sind nicht im Piconet verbunden.

2. Die Station, welche als erste eine Verbindung aufbaut, ist der Master. Die anderen Stationen sind Slaves (kann auch nur ein Slave sein). Der Verbindungsaufbau erfolgt entweder durch das Senden einer Page-Nachricht (MAC Adresse ist bekannt) auf 16 bzw. 32 Frequenzsprüngen oder einer Inquiry (MAC Adresse ist nicht bekannt und wird erst ermittelt) Nachricht an alle Stationen. Die maximale Verzögerung bevor der Master einen Slave erreicht, beträgt 2,56 Sekunden.
3. Danach befindet sich die Station im Status „Verbunden“.
4. Nach Empfang der Nachricht, wird „Detach“ (Verbindung Trennen) gesendet und der Empfänger geht in den „Standby“ Mode.

Um Strom zu sparen, können Stationen die nicht Senden oder Empfangen in verschiedene Zustände wechseln. Diese Zustände sind in der Reihenfolge ihres Energieverbrauchs (ansteigend) aufgelistet:

- **Sniff:** Hierbei wird in einstellbaren periodischen Abständen die Schnittstelle auf Übertragungen überprüft.
- **Hold:** Slaves können in diesen Zustand - vom Master oder sich selbst initiiert - wechseln. Die MAC Adresse bleibt erhalten.
- **Park:** Die Stationen nehmen nicht an der Kommunikation teil. Die MAC Adresse wird zurückgegeben. Der Verkehr des Master wird aber für Synchronisationszwecke aufgenommen. Broadcast Nachrichten (Nachricht an alle) werden erkannt und verarbeitet.

2.3.4 Protokollstack

Hier wollen wir einen kleinen Einblick in den Protokollstack geben, um die verschiedenen Profile und deren Aufbau besser verstehen zu können. Weitere Informationen sind im Bluetooth-Whitepaper [11] zu finden.

Wie das OSI-Modell, greift auch die Bluetooth-Spezifikation für ihre Protokollarchitektur auf ein Schichtenmodell zurück, um die Entwicklung von Anwendungen, die im Rahmen dieser Spezifikation zusammenarbeiten können, zu unterstützen. Es kann sein, dass verschiedene Anwendung auch verschiedene Protokollstacks verwenden, wobei die Protokollstacks immer eine gemeinsame Sicherheits- und Verbindungsschicht verwenden. Nicht alle Protokolle sind spezifische Bluetooth-Protokolle (siehe Tabelle 2.2) und es werden von den Anwendungen auch nicht immer alle Protokolle im Protokollstack benutzt.

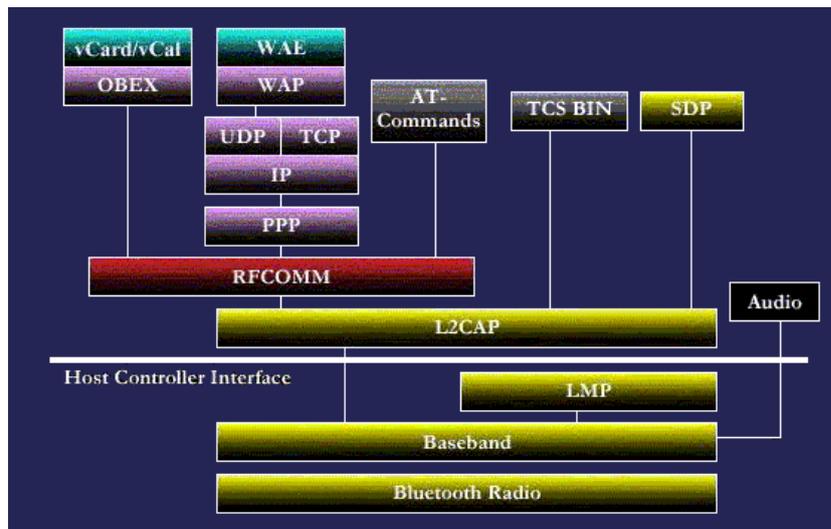


Abbildung 2.6: Bluetooth Specification Protocol Stack. [14]

Die Offenheit der Bluetooth-Spezifikation ermöglicht den Herstellern die freie Implementierung eigener (herstellerspezifischer) Anwendungsprotokolle oberhalb der Bluetooth-spezifischen Protokolle, so dass die Anwendungen, die die Bluetooth Fähigkeiten in vollem Umfang nutzen leicht erweitert werden können. So können z.B. auch zusätzliche Protokolle wie HTTP (Hypertext Transfer Protocol) und FTP (File Transfer Protocol) so eingepasst werden, dass sie oberhalb der für Bluetooth spezifischen Protokolle zusammenarbeiten können. Ausserdem wird hier eine Hostcontroller-Schnittstelle (HCI, Host Controller Interface) definiert, die eine Befehlsschnittstelle zum Basisbandcontroller, Link Manager und zum Zugriff auf den Hardwarestatus und die Steuerregister darstellt. Diese HCI kann ober- oder unterhalb der L2CAP liegen. Der Bluetooth-Protokollstack besteht aus vier Schichten. (siehe Tabelle 2.2) Bei der Kabelersatzschicht, der Telefoneschicht und der Schicht mit den adaptierten Protokollen redet man von den anwendungsbezogenen Protokollen mit denen man die Anwendungen über die Bluetooth-Kernprotokolle ablauffähig bekommt.

Die Bluetooth-Kernprotokolle

Sowohl die für die drahtlose Bluetooth-Technologie zuständigen als auch die RFCOMM und TCS BIN Protokolle sind von der SIG entwickelt worden, wobei letztere jedoch auf vorhandene Standards basieren. Die meisten Bluetooth-Geräte verlangen die Kernprotokolle und das des Bluetooth-Funks (siehe Abbildung: 2.6), während die anderen Protokolle nur bei Bedarf verwendet werden.

Bluetooth-Protokollschicht	Komponenten des Protokollstacks
Bluetooth-Kernprotokolle	Basisband LMP (Link Management Protocol) L2CAP (Logical Link Control and Adaption Protocol) SDP (Service Discovery Protocol)
Kabelersatzprotokoll	RFCOMM (Radio Frequency Communication)
Telefonieprotokolle	TCS BIN (Telefony Control Specification Binary) AT-Befehle
Adaptierte Protokolle	PPP (Point-to-Point Protocol) UDP (User Datagram Protocol) TCP (Transmission Control Protocol) IP (Internet Protocol) OBEX (Object Exchange Protocol) WAP (Wireless Application Protocol) vCard vCalendar IrMC (Infrared Mobile Communications) WAE (Wireless Application Environment)

Tabelle 2.2: Die Protokolle und Schichten des Bluetooth-Protokollstacks

Basisband ermöglicht physikalische Funkverbindung zwischen verschiedenen Geräten im Pico-Netz. Es gibt 2 Arten phys. Verbindungen mit ihren entsprechenden Basispaketen:

1. SCO (Synchronous Connection-Oriented) SCO-Pakete enthalten nur Audio- oder Audio- und Videodaten.
2. ACL (Asynchronous Connection-Less) ACL-Pakete werden nur von Daten verwendet.

Das LMP (Link Manager Protocol) ist für folgende Aufgaben zuständig:

- Aufbau der Verbindung
- Kontrolle zwischen den Bluetooth Geräten.
- Steuerung der Bluetooth Geräte.
- Aushandeln der Basisbandpaketgrößen

- Sicherheit (Authentifizierung und Verschlüsselung, sowie Generieren, Austausch und Überprüfung von Verbindungs- und Verschlüsselungsschlüsseln.)
- Kontrolliert Energiemodi und Betriebszyklen der Bluetooth-Funkeinheit
- Kontrolliert Verbindungsstatus einer Bluetooth-Einheit im Pico-Netz.

Das L2CAP (Logical Link Control and Adaptation Protocol) unterstützt folgende Eigenschaften:

- Multiplexing der höheren Ebene
- Paketsegmentierung
- Paketneuzusammensetzung
- QoS (Quality of Service)

L2CAP erlaubt das Empfangen/Senden von Paket bis 64 KB für die Protokolle der höheren Ebene. Das L2CAP Protokoll sieht nur ACL-Verbindungen vor, obwohl das Basisprotokoll die zwei Verbindungstypen: ACL und SCO vorsieht. Audiodaten können jedoch auch verpackt und mit dem Kommunikationsprotokollen über L2CAP gesendet werden.

Das SDP (Service Discovery Protocol) ist ein Protokoll zur Diensterkennung, zur Abfrage von Leistungsmerkmalen aller Dienste und zur Abfrage von Geräteinformationen. Ein Benutzer kann Dienste nutzen, die in der Umgebung gefunden wurden. Anschließend kann eine Verbindung zwischen zwei oder mehreren Geräten aufgebaut werden.

Kabelersatzprotokoll stellt in der Bluetooth-Spezifikation ein wichtiges serielles Emulationsprotokoll da, um Daten ohne Kabel über ein seriellen Anschluss zu übertragen.

RFCOMM (Radio Frequency Communication)

- Serial Line Emulation Protocol, basiert auf ETSI 07.10 Spezifikation
- emuliert das RS-232 Protokoll
- unterstützt viele Anwendungen, die den seriellen Anschluss benutzen, wie z.B. PPP oder OBEX

Telefonieprotokoll

- **TCS BIN (Telephony Control Specification Binary)**
Bitorientiertes Protokoll, enthält Rufkontrolle, Verbindungsaufbau, Sprachübertragung, Datenübertragung.

- **AT-Befehle**

werden über RFCOMM gesendet, zur Steuerung von Handys und Modems. Auch Befehle für eine Art FAX-Übertragung sind möglich.

Adaptierte Protokolle

Hierbei handelt es sich um bereits existierende Protokolle, die für verschiedene Zwecke in den höheren Schichten verwendet werden. Diese ermöglichen es älteren Anwendungen mit der drahtlosen Bluetooth-Technologie zusammenzuarbeiten.

PPP

Point to Point Protokoll setzt auf RFCOMM auf. Ist für die Tunnelung von IP Paketen gedacht (wie auch bei Modems üblich).

TCP/UDP/IP

Diese Internetprotokolle ermöglichen die Einbindung von Bluetooth-Geräten in LANs oder WANs. Es ist das weitverbreitetste Protokoll überhaupt und wird oft in Verbindung mit PPP verwendet.

Es gibt noch weitere adaptierte Protokolle (siehe Tabelle 2.2) die wir hier nicht weiter erläutern möchten.

2.3.5 Profile

Die Interoperabilität zwischen Bluetooth Devices verschiedener Hersteller wird durch sogenannte Profile gewährleistet, die für bestimmte Anwendungsbereiche festgelegt sind. Wenn eine Bluetooth Verbindung aufgebaut wird, tauschen die Systeme ihre Profile aus und legen damit fest, welche Dienste sie für die jeweiligen anderen Partner zur Verfügung stellen können und welche Daten oder Befehle sie dazu benötigen. Das Headset Profil fordert beispielsweise von einem Bluetooth kompatiblen Mobiltelefon einen Audiokanal an und steuert über zusätzliche Datenkanäle die Lautstärkeregelung.

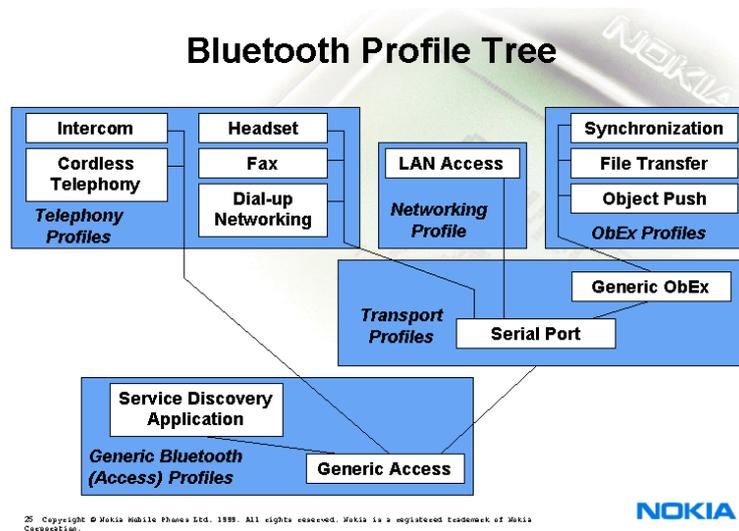


Abbildung 2.7: Übersicht der Profile und deren Abhängigkeiten voneinander [13]

Alle Bluetooth Produkte müssen das Generic Access Profile (GAP) unterstützen, damit die Kommunikation zwischen zwei oder mehr Geräten überhaupt aufgebaut werden kann. Darüber hinaus haben bestimmte Profile andere Grundprofile als Voraussetzung.

- Um ein Fax-Profil starten zu können, muß z.B. zunächst das Serial Port Profil und das GAP aktiviert sein.
- Zum Übertragen von Dateien über das File Transfer Profile sind das Generic Object Exchange Profile, das Serial Port Profile und das GAP notwendig.

Ein Übersicht der Profile und deren Abhängigkeiten voneinander sind in Abbildung 2.7 zu sehen.

Die folgenden Profile sind aus dem Whitepaper 1.1 [11] entnommen:

GAP - Generic Access Profile:

- Ist das allgemeines Zugangsprofil.
- Definiert allgemeine Prozeduren zur Erkennung von Bluetooth-Geräten sowie Prozeduren für das Verbindungsmanagement (Link Management).
- Beschreibt den Verbindungsaufbau und -status, das Verhalten im Standby Modus, regelt die Verarbeitung weiterer spezifischer Profile und die Sicherheitsprozeduren (z.B: Austausch PIN)

SDAP - Service Discovery Application Profile:

Legt fest, auf welche Weise ein Bluetooth Device nach dem gewünschten Service sucht. Wird z.B. eine Audioübertragung angefordert, dann sucht das SDA Profil nach allen Bluetooth Komponenten in Reichweite, die diesen Dienst anbieten könnten und stellt eine Verbindung her, die dann von der Anwendung genutzt werden kann.

CTP - Cordless Telephony Profile:

Z.B. für Audioübertragung, ein- und abgehende Gespräche, Konferenzschaltung und Anrufer-ID (ISDN).

IntP - Intercom Profile:

Definiert, teilweise in Ergänzung des Telephony Protokolls, wie zwei Bluetooth Geräte im Intercom Modus gesteuert werden, z.B. GSM-Mobiltelefone, die ohne Serviceprovider als Walkie-Talkies (Sprechfunkgeräte) genutzt werden sollen.

SPP - Serial Port Profile:

Emuliert ein serielles Kabel (RS-232). Hierzu wird das RFCOMM-Protokoll verwendet.

HSP - Headset Profile:

Legt die Funktionen für ein Headset oder eine Freisprecheinrichtung fest, z.B. Audiokanäle und Lautstärkeregelung.

DUNP - Dial-Up Networking Profile:

Z.B. für Internet-Zugang oder Remote-Access-Anwendungen, Anwahlprozeduren, AT-Modem-Befehle und Wahlverfahren.

FaxP - Fax Profile:

Senden und Empfangen von Faxen, Weiterleitung der Faxdatei z.B. vom Mobiltelefon an ein Notebook.

LAP - LAN Access Profile:

LAN Zugriffe über PPP, IP Protokoll, PC zu PC Verbindungen (Peer-to-Peer) und NetBIOS Unterstützung

GOEP - Generic Object Exchange Profile:

Das generelle Profil, für den Austausch von Daten-Objekten, z.B. Filetransfer und Synchronisation

OPP - Objekt Push Profile:

Mit diesem Profil sendet ein Bluetooth Gerät ein (Daten-) Objekt an ein anderes Gerät z.B. an ein Notebook, der einen Termin in die Inbox eines Mobiltelefons stellt.

FTP - File Transfer Profile:

Vorgänge, bei denen Dateien und Ordner über Bluetooth zwischen zwei Systemen (Notebooks, PDAs) ausgetauscht werden.

SP - Synchronization Profile:

(Automatische) Synchronisation zwischen Systemen (PCs, Notebooks, PDAs) und deren Applikationen (Mail, Kalender, Datenbanken).

2.3.6 Architektur

Das Bluetooth-System besteht aus 3 Hauptkomponenten:

1. **Radio:** besteht aus einem Sende- und Empfangsmodul und weiterer analoger Radioelektronik.
2. **Link- oder Baseband-Controller (LC):** ist vergleichbar mit der MAC- und LLC - Schicht im ISO/OSI - Schichtenmodell. D.h. hier wird die Fehlerbehandlung, Kommunikationsaufbau, Verbindungsverwaltung, Datensicherheit, Datenkompression, Audiokommunikation, Hopping - Algorithmus, Identifikation und Zugriff kontrolliert und gesteuert.
3. **Link Manager (LM):** bereitet die Daten für die Endgeräte (Hosts) auf und stellt dann die Verbindung zu diesen her.

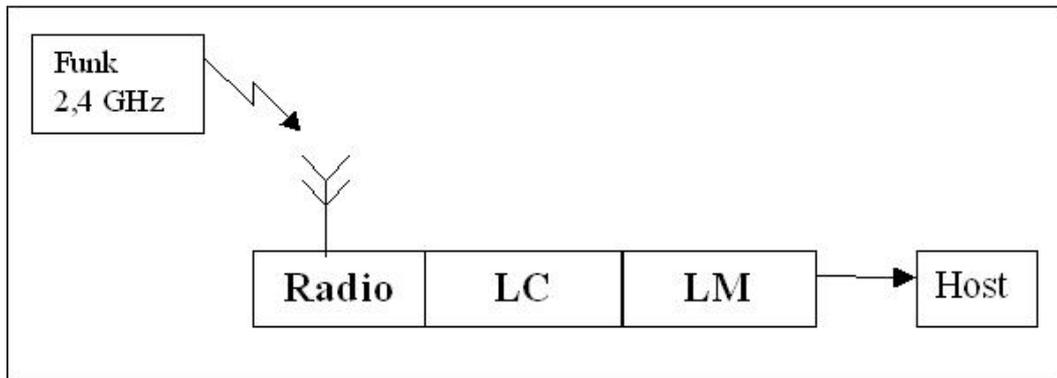


Abbildung 2.8: Grobe Ansicht des Bluetooth - Moduls

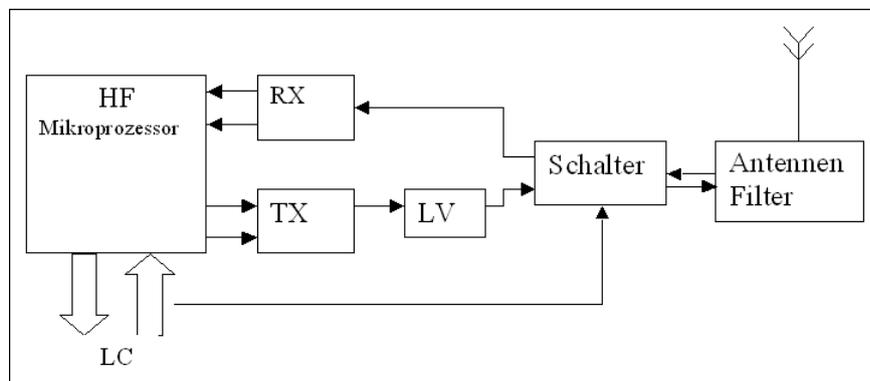


Abbildung 2.9: Radio - Komponente

HF:	Hochfrequenz-Mikroprozessor zur Aufbereitung der schwachen und fehlerhaften Analogsignale
RX:	Empfangeinheit
TX:	Sendeeinheit
LV:	Leistungsverstärker (von 10mW auf 100mW)
LC:	Verbindung zum Link-Controller
Schalter:	Reglung von Empfang und Senden über LC

Tabelle 2.3: Abkürzungen für die Radio - Komponente

Kapitel 3

Analyse

In diesem Kapitel wird näher auf die Beschreibung von 1.3 eingegangen. Da die Diplomarbeit aus drei verschiedenen Anwendungsprogrammen besteht, wird das Kapitel Analyse in drei Abschnitte aufgeteilt:

- **Kapitel 3.1**
Analyse des PES (Präsentations-Erstellungssystem)
- **Kapitel 3.2**
Analyse der PSE (Präsentations-Steuereinheit)
- **Kapitel 3.3**
Analyse der PAE - Präsentations-Anzeigeeinheit

In diesen Abschnitten wollen wir auf folgende Punkte eingehen:

- Anforderungsanalyse
 1. Festlegung der Anforderungen
 2. Ermittlung und Spezifikation der Anwendungsfälle auf Basis der Anforderungen
 3. Grafische Benutzeroberflächen

3.1 PES - Präsentations-Erstellungssystem

3.1.1 Festlegung der Anforderungen

1. Erstellung einer Präsentation
Zunächst müssen die Präsentationsfolien mit einem Anwendungsprogramm erstellt werden.
2. Speichern und Senden der Präsentation
Die Präsentation wird gespeichert und vom PES zur PSE übertragen.

3.1.2 Ermittlung der Anwendungsfälle auf Basis der Anforderungen

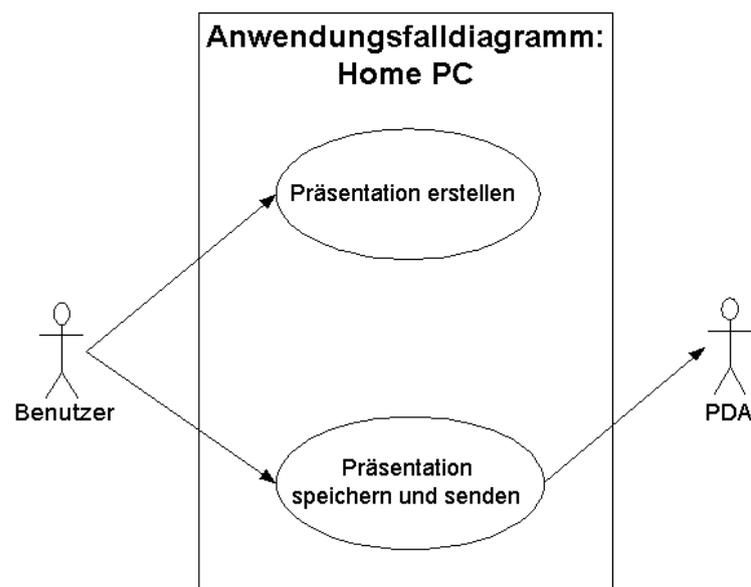


Abbildung 3.1: Anwendungsfall PES

3.1.3 Spezifikation der Anwendungsfälle

Eine zusätzliche Spezifikation der Anwendungsfälle ist hier nicht nötig, da alle wichtigen Details schon in den Anforderungen beschrieben wurden. (siehe Unterkapitel 3.1.1 Festlegung der Anforderungen)

3.2 PSE - Präsentations-Steuereinheit

3.2.1 Festlegung der Anforderungen

1. PAE auswählen
Es soll zunächst nach Bluetoothfähigen Geräten im Umfeld gesucht werden und danach eine PAE aus der Liste der gefundenen Geräte auswählbar sein. Befindet sich kein Gerät im Umfeld, soll es möglich sein, die Präsentation ohne Verbindung zur PAE auf der PSE zu starten. Hierbei sollen die Folien nur auf der PSE angezeigt werden.
2. Präsentation auswählen
Damit auch mehrere Präsentationen auf der PSE vorhanden sein können, soll eine Liste zur Auswahl angezeigt werden, aus der dann eine Präsentation ausgewählt werden kann.
3. Anzahl der übertragenden Folien anzeigen
Die Anzahl der bereits übertragenen Folien zur PAE soll auf dem Display angezeigt werden. (Z.B.: Übertragene Folien zum Beamer: 5 von 25)
4. Anzeige der aktuellen Foliennummer
Die Nummer der angezeigten Folie und die Gesamtzahl des Foliensatzes soll auf dem Display angezeigt werden.(z.B.: akt. Folie 5 von 25)
5. Vor- und zurückblättern der Folien (sequentiell)
Die Folien sollen auf der PSE vor- und zurück geblättert werden können.
6. Eine bestimmte Folie auswählen (selektiv)
Eine bestimmte Folie soll aus einer Liste, aller Folien eines Foliensatzes, ausgewählt werden können.
7. Diashow
Es sollen einzelne Folien aus dem Foliensatz auswählbar und deren Reihenfolge änderbar sein. Folien sollen automatisch nach bestimmten Zeitabständen vorgeblättert werden können. Die Zeitabstände sollen änderbar sein.
8. Diashow-Profil speichern und laden
Die Auswahl der Folien sowie deren Reihenfolge und deren Anzeigedauer sollen in einem Profil abspeicherbar sein.
9. Anzeigebestätigung für einzelne Folien
Um die Folien auf der PSE auch auf der PAE sichtbar zu machen, soll eine Anzeigebestätigung vom Anwender betätigt werden. Die Anzeigebestätigung soll auch deaktivierbar sein, so dass die auf der PSE zu sehende Folie auch sofort auf der PAE angezeigt wird.

3.2.2 Ermittlung der Anwendungsfälle auf Basis der Anforderungen

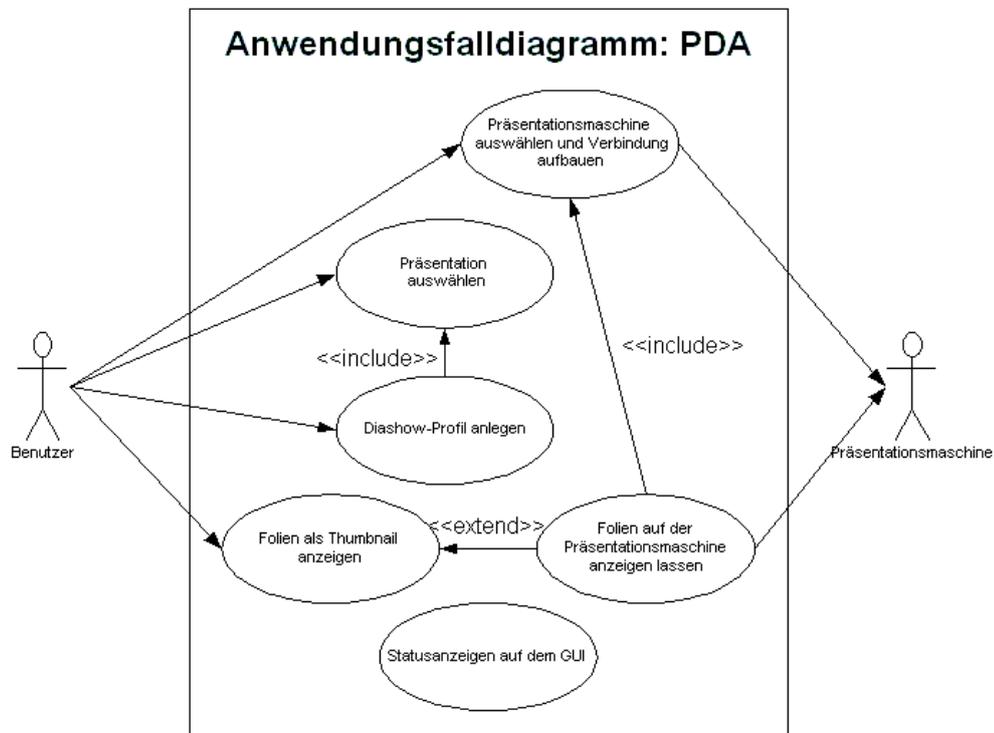


Abbildung 3.2: Anwendungsfall PDA

3.2.3 Spezifikation der Anwendungsfälle

Als erstes werden die Interaktionen zwischen dem Anwender und den Anwendungsfällen spezifiziert. Da es mehrere Varianten der Präsentation gibt, ist es sinnvoll sie in einzelnen Szenarien näher zu beschreiben.

Möglichen Varianten sind:

- **Sequentiell** (Folien werden nacheinander angezeigt)
Diese Variante ist für eine Person gedacht, die speziell einen Foliensatz erstellt hat, den er sehr gut kennt und nur der Reihe nach, Folie für Folie, anzeigen möchte.
- **Selektiv** (Folien können einzeln ausgewählt werden)
Diese Variante macht z.B. Sinn, wenn es einen grossen Foliensatz gibt der von mehreren Personen verwendet wird. Jede Person braucht nur eine bestimmte Auswahl von Folien aus diesem Foliensatz (z.B.: Folie 5-10,14,45...), so dass eine selektive Auswahl benötigt wird, da eine Sequentielle Auswahl sehr lange dauern würde.

- Diashow (Folien werden automatisch in bestimmten Zeitabständen nacheinander angezeigt)

Wenn die Anzeigedauer der einzelnen Folien vorher feststeht oder die gesamte Präsentation zeitlich begrenzt ist, ist diese Variante sehr sinnvoll. Die Diashow soll hauptsächlich dann eingesetzt werden, wenn der visuellen Präsentation mehr Bedeutung beigemessen wird, als der verbalen Präsentation, bzw. wenn der Vortragende genau weiß, wieviel Redezeit er für die einzelnen Folien benötigt. Diese Variante wäre zum Beispiel in Form einer Messe-Präsentation, die tonlos als Diashow ablaufen würde, denkbar. Hierbei wäre es möglich, aus einem bestehenden Foliensatz bestimmte Folien auszuwählen und deren Anzeigedauer einzeln einzustellen.

Jetzt muss noch unterschieden werden, ob die Präsentation nur auf der PSE oder auch auf der PAE angezeigt werden soll. Beides ist sinnvoll, z.B. während der Bahnfahrt hat der Anwender nochmal die Möglichkeit sich einen Überblick über seinen Foliensatz zu verschaffen. In diesem Beispiel werden die Folien nur auf der PSE und nicht auf der PAE angezeigt. Desweiteren ist es auch möglich die Präsentation gleichzeitig auf der PSE und der PAE anzeigen zu lassen. Dies ist sinnvoll, da der Vortragende während der Präsentation in Richtung Zuhörer sehen möchte und gleichzeitig die aktuell auf der PAE angezeigte Folie als Thumbnail¹ auf dem Display der PSE sehen kann. Ausserdem hat der Anwender die Möglichkeit, die Folien auf der PSE auszuwählen, bevor er sie den Zuhörern auf der PAE sichtbar macht.

Aus diesen Gründen wurden folgende Szenarien ausgewählt:

1. Szenario: Präsentation auswählen

Nachdem das Programm gestartet wurde, erscheint eine „Look & Feel“-Oberfläche² auf dem Display, über die es möglich ist einen Foliensatz zu öffnen.

- (a) Der Anwender wählt aus einer Liste von Foliensätzen einen Foliensatz aus und öffnet ihn

Hiermit ist der Foliensatz ausgewählt.

2. Szenario: PAE auswählen

Nachdem das Programm gestartet wurde, wird zunächst nach Bluetoothfähigen Geräten im Umfeld gesucht.

- (a) Der Anwender wählt aus einer Liste von Bluetooth-Geräten aus seiner Umgebung das Gerät aus, auf dem er die Präsentation anzeigen möchte.

¹Engl. für Daumennagel. Bezeichnung für kleine Bilddarstellungen. Hier wird der Begriff verwendet, da man auf der PSE nur eine kleine Version der Originalfolie darstellen kann.

²Look & Feel bedeutet hier das der Anwender einen optisch vertraute Oberfläche vorfindet (im Stil von PocketPC 2002)

- (b) Nun wird eine Verbindung zwischen der PSE und der ausgewählten PAE hergestellt.

Nachdem der Foliensatz und die PAE ausgewählt wurden, kann mit der Präsentation begonnen werden.

3. Szenario: Präsentation als Diashow erstellen und als Profil speichern

Nachdem der Foliensatz ausgewählt wurde startet der Anwender den Präsentations-Modus ohne PAE. Die erste Folie wird auf dem Display der PSE angezeigt.

- (a) Der Anwender startet die Diashow
- (b) Jetzt wählt der Anwender die einzelnen Folien aus, die er für die Diashow verwenden möchte und legt deren Reihenfolge fest.
- (c) Danach kann der Anwender die Anzeigedauer der einzelnen Folie in Sekunden einstellen.
- (d) Nachdem er die Einstellungen vorgenommen hat, speichert er diese als Profil ab.

Der Anwender kann beim Start der Diashow-Präsentation das vorher erstellte und abgespeicherte Profil laden und sofort mit der Präsentation beginnen. Dabei wird Folie für Folie auf der PSE und auf der PAE mit der eingestellten Anzeigedauer automatisch angezeigt.

4. Szenario: Präsentation mit selektiver Auswahl ohne PAE

Nachdem der Foliensatz ausgewählt wurde und kein Bluetooth-Gerät im Umfeld der PSE gefunden wurde, kann der Anwender nur den Präsentations-Modus ohne PAE starten. Die erste Folie wird auf dem Display der PSE angezeigt.

- (a) Der Anwender läßt sich die Liste der Folien der Präsentation anzeigen.
- (b) Aus dieser Liste wählt er eine bestimmte Folie aus.
- (c) Die ausgewählte Folie wird auf dem Display der PSE angezeigt.

Dieses kann beliebig oft wiederholt werden.

5. Szenario: Präsentation mit sequentieller Auswahl mit Anzeige auf einer PAE

Nachdem der Foliensatz ausgewählt wurde, wählt der Anwender eine PAE aus einer Liste der vorhandenen Bluetooth-Geräte im Umfeld aus. Danach wird der Präsentations-Modus gestartet und die erste Folie wird auf dem Display der PSE angezeigt.

- (a) Der Anwender macht jetzt die auf der PSE angezeigte Folie, durch eine Bestätigung, auf der PAE sichtbar.
- (b) Der Anwender wählt die nächste Folie aus, die auf dem Display der PSE angezeigt wird.

- (c) Durch eine Bestätigung macht der Anwender diese Folie jetzt auf der PAE sichtbar.
- (d) Der Anwender schaltet die Anzeigebestätigung aus.
- (e) Durch Zurückblättern wählt der Anwender die vorherige Folie aus dem Foliensatz aus, die auf dem Display der PSE und auch gleich auf der PAE angezeigt wird.

Das Vor- und Zurückblättern der Folien läuft hier sequentiell ab. Um die Folien, die auf dem Display der PSE angezeigt werden, auch auf der PAE sichtbar zu machen, muss der Anwender dieses jeweils einzeln bestätigen, es sei denn, er hat die Anzeigebestätigung deaktiviert.

3.2.4 Mögliche grafische Benutzerschnittstellen für die PSE

Für die Darstellung der grafischen Benutzerschnittstellen auf der PSE wird hier die Oberfläche eines PDAs benutzt. Die Screenshots sind in diesem Fall mit dem Emulator des Embedded Visual Tool 3.0 von Microsoft erstellt worden.



Abbildung 3.3: Vorauswahlfenster auf dem PDA

Nach dem Programstart soll ein Vorauswahlfenster auf dem Display des PDAs zu sehen sein, auf dem man den Foliensatz und die PAE auswählen kann. (siehe Abbildung 3.3)

Nachdem die Vorauswahl getroffen wurde, erscheint als nächstes das Hauptfenster (siehe Abbildung 3.4) auf dem die erste Folie des ausgewählten Foliensatzes



Abbildung 3.4: Hauptfenster auf den PDA

zu sehen sein soll (Folienbild auf der Abbildung noch nicht sichtbar). Von diesem Fenster aus soll es möglich sein, die Präsentation sequentiell, selektiv und im Diashow-Modus zu bedienen.



Abbildung 3.5: Die Diashow wird im Hauptfenster des PDA selektiert

Vom Hauptfenster aus soll es möglich sein, in den Diashow-Modus zu wechseln (siehe Abbildung 3.5)

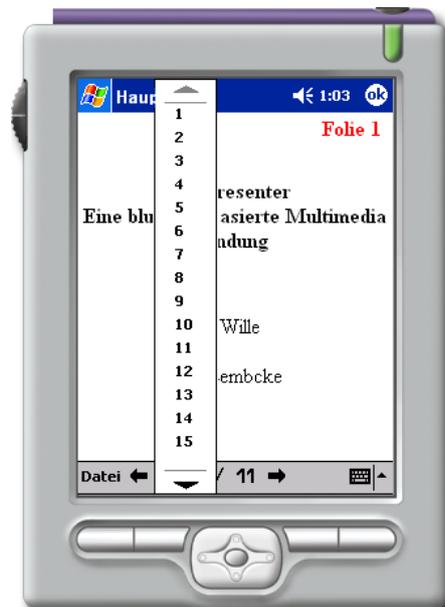


Abbildung 3.6: Eine Folie wird im Hauptfenster des PDA ausgewählt

Ausserdem soll man aus dem Hauptfenster heraus auch die selektive Auswahl bedienen können. (siehe Abbildung 3.6)

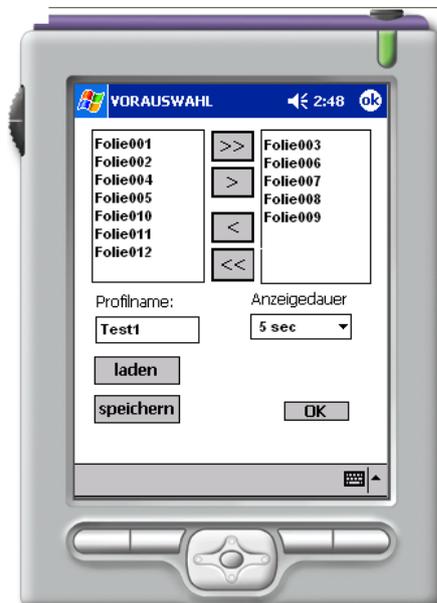


Abbildung 3.7: Diashowauswahl auf dem PDA

Im Diashow-Modus (siehe Abbildung 3.7) sollen Einstellungen wie Folienauswahl, -reihenfolge und -anzeigedauer möglich sein. Desweiteren soll es auch möglich sein Profile zu speichern und wieder zu laden.

3.3 PAE - Präsentations-Anzeigeeinheit

3.3.1 Festlegung der Anforderungen

1. Verbindungsaufbau mit dem PDA
Es soll zunächst eine Verbindung zwischen PDA und PAE aufgebaut werden.
2. Eine bestimmte Folie auf dem Anzeigerät anzeigen lassen
Eine bestimmte Folie soll auf dem Anzeigerät sichtbar werden, nachdem der Anwender dieses auf dem PDA bestätigt hat.

3.3.2 Ermittlung der Anwendungsfälle auf Basis der Anforderungen

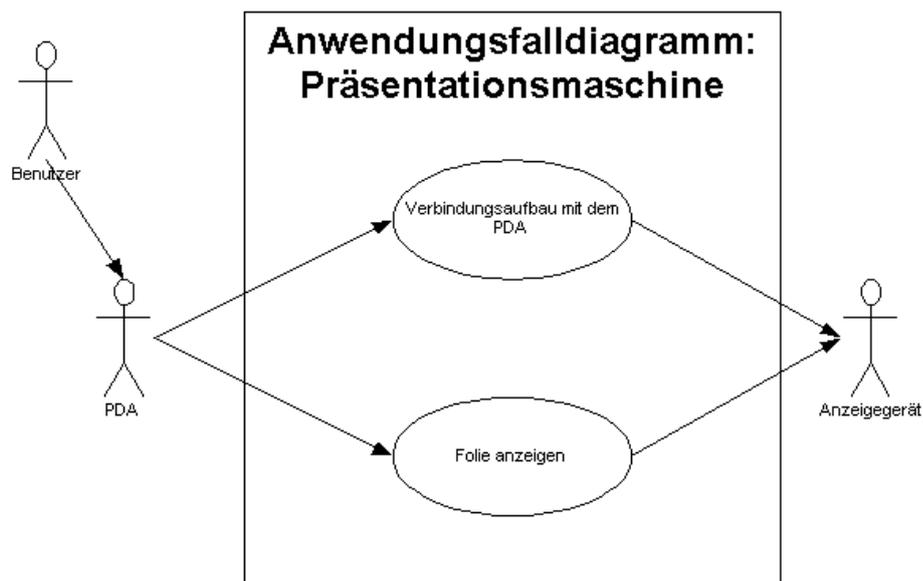


Abbildung 3.8: Anwendungsfall PAE

3.3.3 Spezifikation der Anwendungsfälle

Eine zusätzliche Spezifikation der Anwendungsfälle ist hier nicht nötig, da alle wichtigen Details schon in den Anforderungen beschrieben wurden. (siehe Unterkapitel 3.3.1 Festlegung der Anforderungen)

Zum Abschluß wird noch mal ein Sequenzdiagramm gezeigt, welches das Gesamtsystem darstellen soll.

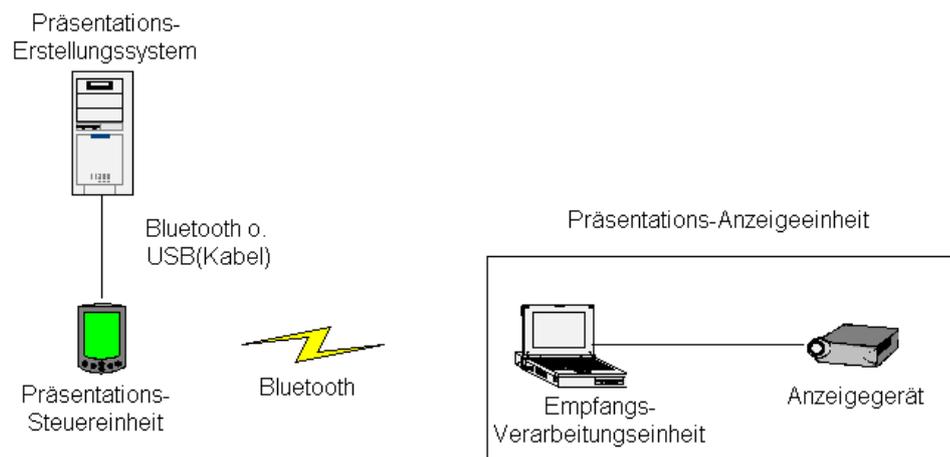


Abbildung 3.9: Gesamtüberblick der Systemarchitektur - 2. Version

Kapitel 4

Das Labor

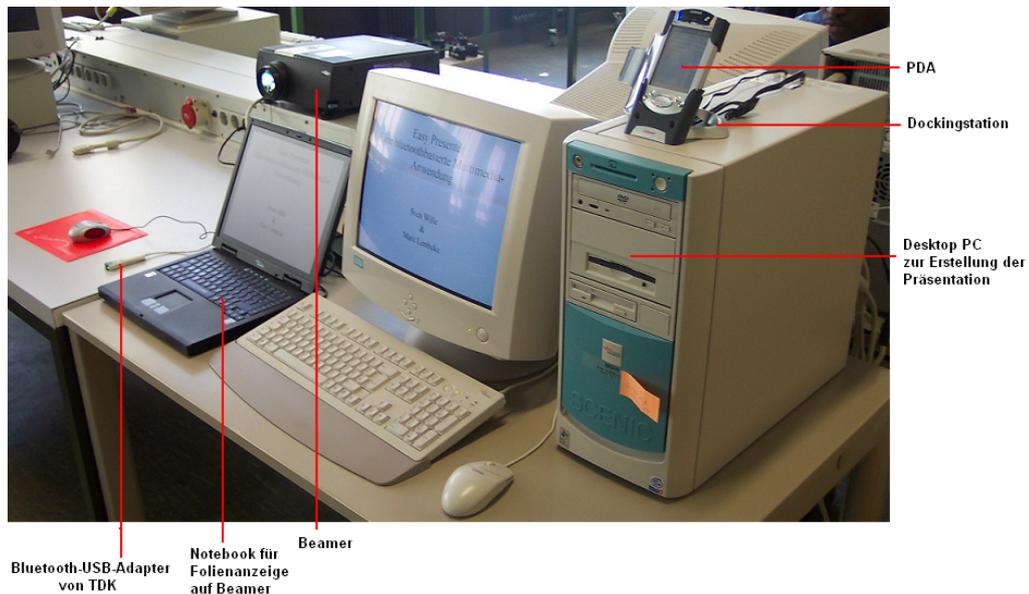


Abbildung 4.1: Gesamtübersicht des Labors

In diesem Kapitel werden die Hardwarekomponenten: Desktop PC, PDA, Präsentationsmaschine und deren Kommunikationsschnittstellen¹, so wie sie in unserer Diplomarbeit verwendet werden, näher beschrieben (siehe Abbildung 4.1 Gesamtübersicht des Labors). Desweiteren werden hier alle zusätzlichen „Werkzeuge“, die wir für unsere Diplomarbeit benötigen, aufgelistet und erklärt. Mit Werkzeugen sind z.B.: Bibliotheken, USB-Bluetooth-Adapter, Entwicklungsumgebungen, Anwendungsprogramme usw. gemeint.

¹Kommunikationsschnittstelle I: zwischen Präsentations-Erstellungssystem und PDA
Kommunikationsschnittstelle II: zwischen PDA und Präsentationsmaschine

4.1 Desktop PC

Das Präsentations-Erstellungssystem ist das Rechnersystem, auf dem die Präsentation erstellt und gespeichert wird. In unserem Fall benutzen wir zur Erstellung der Präsentation ein handelsüblichen PC, im weiteren Verlauf der Diplomarbeit wird dieser als Desktop PC bezeichnet.

4.1.1 Technischen Daten

Prozessor:	1600 MHz Intel Pentium 4 Prozessor
Speicher:	512 MB RAM
Festplatte:	20 GB
Anschlüsse:	Serieller/USB-Anschluß
Betriebssystem:	Windows XP Professional (deutsch)

Tabelle 4.1: Technische Daten Desktop PC

4.1.2 Kommunikationsschnittstelle zum PDA

Es müssen die Folien über die Kommunikationsschnittstelle I zum PDA übertragen werden können. Hierfür gibt es zwei verschiedenen Möglichkeiten. Zum einen über die USB-Docking-Station oder alternativ über den Bluetooth-USB-Adapter von TDK.

USB-Docking-Station für den PDA

Die Docking-Station kann einerseits zum Laden des PDAs genutzt werden, indem man das mitgelieferte Netzteil an die Docking-Station anschließt und den PDA in die Docking-Station schiebt. Andererseits kann die Docking-Station auch zur Synchronisation des PDAs mit dem PC genutzt werden, z.B. über ActiveSync. Wie ActiveSync funktioniert wird im weiteren Verlauf der Diplomarbeit noch beschrieben. Die USB-Docking-Station kann hierfür entweder über das USB-Kabel oder über das serielle Kabel an den Desktop PC angeschlossen werden.

TDK Bluetooth-USB-Adapter

Mit Hilfe des Bluetooth-USB-Adapters ist es möglich, den PDA drahtlos mit dem PC zu verbinden. Dazu wird der Adapter an die USB-Schnittstelle des PCs angeschlossen. Folgend noch einige für uns wichtige Daten zum Bluetooth-USB-Adapter [17]:

- Bluetooth-Spezifikation: Kompatibel zu V 1.1
- Datenübertragungsrate (tatsächlich): 721 kbps (max)
- Profile: Allgemeiner Zugriff, Diensterkennung, DFÜ-Netzwerk, Dateiübertragung, Serieller Anschluss, OBEX, LAN-Zugriff, Fax, Synchronisierung
- Funkübertragungsstärke: Leistungsklasse 2: 0 dBm (1 mW) 0 dBm (typisch) & 4 dBm (max)

4.1.3 Softwarekomponenten

Um die Präsentation erstellen und zum PDA übertragen zu können, werden noch einige Softwarekomponenten benötigt. Unter anderem ein Betriebssystem, ein Programm zur Erstellung der Präsentation, ein Programm um die Präsentation zum PDA zu übertragen und die entsprechenden Treiber.

Windows XP

Für unsere Laborumgebung benutzen wir Windows XP Professional Version 2002 mit Service Pack 1.

Powerpoint

Zur Erstellung und Speicherung der Präsentationen benutzen wir Powerpoint 2000. Hierbei ist es möglich, die Präsentationen in verschiedene Formate abzuspeichern, z.B. in PPT, HTML und auch in JPG.

Microsoft ActiveSync

Microsoft ActiveSync ist eine Synchronisierungssoftware für Pocket PCs mit Windows CE. Hier wird es verwendet, um Daten zwischen zwei Bluetooth-Geräten (PDA und PC) auszutauschen. Wir benutzen die Version ActiveSync 3.5.

Treiber

Bluetooth Driver Win 98 SE/Me/2000/XP(Dec 2002)

4.2 PDA

4.2.1 Technische Daten

Prozessor:	206 MHz Intel StrongARM SA-1100 32-bit RISC Prozessor
Speicher:	32-MB SDRAM, 16-MB Flash-ROM
Anschlüsse:	Serieller/USB-Anschluß, Infrarot-Schnittstelle (115 Kbps)
Anzeige:	12-bit (4096 Farben) TFT (thin film transistor) LCD-Display mit einer Auflösung von 240 x 320 Bildpunkten (Touchscreen)
Zusätzlich:	für das Senden und Empfangen über Funk ist ein Bluetooth Modul integriert

Tabelle 4.2: Technische Daten PDA: *iPAQ H3870*

4.2.2 Betriebssysteme

Windows CE

Windows CE ist ein von Microsoft entwickeltes modulares 32 Bit-Betriebssystem für den Einsatz in Embedded-Systems und einer Vielzahl von Geräten, z.B. ein PDA zur Kommunikation und mobilen Datenverarbeitung.

Der wesentliche Unterschied zu anderen Betriebssystemen der Windows-Familie ist, dass geringe Hardwareressourcen benötigt werden. Für den Betrieb von Windows CE in der Minimalkonfiguration sind nur 28kB RAM und 330kB ROM notwendig.

Von Windows CE werden folgende Prozessortypen unterstützt:

- x86 (486 und höher)
- MIPS

- ARM / StrongArm
- Motorola PowerPC
- Hitachi Super-H (SH3/SH4)

Unser PDA H3870 benutzt den Prozessortyp: ARM / StrongArm

Diese Daten und weiterer Informationen über das Betriebssystem Windows CE sind in der Projekt- und Seminararbeit von Harald Karner [6] zu finden.

4.2.3 Bibliotheken

BTAccess

BTAccess ist eine C++ Bluetooth Klassen Bibliothek für den iPAQ Pocket PC. Diese Bibliotheken bieten ein sehr einfaches Klassenmodell, um auf den Bluetooth Protokollstack, der für die 3870/3970 Serie des iPAQ verwendet wird, zuzugreifen.

<i>BTAccess.h:</i>	Header file for both BTAccess classes <i>CBtStack</i> and <i>CBtDevice</i>
<i>BTAccessDefs.h:</i>	Header file containing various enumerated types and data structures
<i>BTAccess.dll:</i>	DLL containing all BTAccess functionality
<i>BTAccess.lib:</i>	Import library for BTAccess

Tabelle 4.3: Das BTAccess Paket

Es gibt zwei Klassen für BTAccess:

1. CBtStack
2. CBtDevice

Die erste Klasse: *CBtStack* ist für die Definition der Methoden und deren Daten des allgemeinen Stacks, sowie der Geräte-Operationen auf dem iPAQ, verantwortlich.

Die zweite Klasse: *CBtDevice* ist für die Definition der Methoden und deren Daten für die Kommunikation zwischen speziellen Bluetooth-Geräten verantwortlich.

Die Methoden der beiden Klassen sind im Anhang aufgelistet und erklärt.

4.3 Präsentationsmaschine

Für unsere Präsentationsmaschine benötigen wir 2 Geräte:

- Notebook
- Beamer

4.3.1 Notebook

Technische Daten

Prozessor:	Mobile CPU 1.80 GHz Intel(R) Pentium(R)
Speicher:	224 MB RAM
Anschlüsse:	Serieller/USB-Anschluß, Infrarot-Schnittstelle
Zusätzlich:	für Funkempfang und -senden ist ein TDK Bluetooth-USB-Adapter angeschlossen

Tabelle 4.4: Technische Daten Notebook

Kommunikationsschnittstelle II

TDK Bluetooth Adapter

Am Notebook ist ein TDK Bluetooth Adapter über USB angeschlossen, um mit dem PDA kommunizieren zu können. Der Adapter wurde, im Unterabschnitt: 4.1.2 Kommunikationsschnittstelle zum PDA , beschrieben.

Betriebssystem

Microsoft Windows XP

- Home Edition
- Version 2002
- ServicePack 1

4.3.2 Beamer Epson EMP

Als Beamer wird ein in der HAW-Hamburg zur Verfügung stehendes Gerät verwendet.

Technische Daten

Das Gerät ist von Epson aus der Serie: EMP-5350/-7250/-7350

Auflösung:	1024 x 768
Farbwiedergabe:	24 bit, 16,7 Millionen Farben
PC-Eingänge:	2 x RGB D-Sub 15
PC-/Video-RGB-Eingang:	5 x BNC
S-VHS-Video-Eingang:	1 x 4 Pin Mini DIN

Tabelle 4.5: Technische Daten Beamer

Folien, die von der Präsentationsmaschine aufgerufen werden, werden über das Anzeigergerät (Beamer) sichtbar. Die Verbindung zwischen dem Notebook und dem Beamer wird über ein VGA-Kabel hergestellt.

4.4 Erstellungsprogramme

4.4.1 Entwicklungsumgebung für Das Notebook

MS Visual C++ 6.0

Um das Server-Anwendungsprogramm für die Präsentationsmaschine, auf dem Notebook, zu entwickeln, benutzen wir MS Visual C++ 6.0.

4.4.2 Entwicklungsumgebung für den PDA

Embedded Visual C++ 3.0

eMbedded Visual Tools 3.0 - 2002 Edition

Die Microsoft eMbedded Visual Tools 3.0 2002 Edition liefert eine komplette Desktop-Entwicklungsumgebung zur Erzeugung von Anwendungsprogrammen, z.B. für den Pocket

File Name:	evt2002web_min.exe
Download Size:	210485 KB
Date Published:	1/6/2003
Version:	3.0

Tabelle 4.6: Embedded Visual C++ für Windows CE

PC. In EVT 3.0 sind: Microsoft eMbedded Visual C++ 3.0 und Microsoft eMbedded Visual Basic 3.0 enthalten. Desweiteren sind die Installationssätze für die SDKs²: Pocket PC 2002 und Smartphone 2002 vorhanden. Der Emulator für Pocket PC 2002 wird mit der SDK geliefert. Somit ist es möglich, Anwendungsprogramme auf dem Emulator zu testen, ohne das der iPAQ H3870 benötigt wird. [16]

4.4.3 CASE-Tool Together 6.0

Together ist ein visuelles UML Modellierungswerkzeug für Softwareentwicklung in Java, C++ und weiteren Sprachen. Es arbeitet direkt mit dem Sourcecode und kann dadurch ideal skaliert und unmittelbar in existierende Entwicklungsumgebungen und Versionskontrollsystemen integriert werden. Diagramme und Sourcecode werden automatisch miteinander synchronisiert, so daß Veränderungen des Sourcecodes unmittelbar in den Class- und Package-Diagrammen erscheinen und umgekehrt.

Together wurde in dieser Arbeit dazu verwendet, die Sequenz- und Klassendiagramme zu erstellen. Desweiteren wurde damit der erste C++ Code erstellt.

4.4.4 UML Modellierungsprogramm MS Visio 2000

Microsoft Visio 2000 Enterprise Edition ist eine Komplettlösung für automatisierte IT-Entwicklung und -Dokumentation. Hervorragend geeignet ist diese Software zur Erstellung von Website-Übersichten und Datenflussdiagrammen. Ferner ermöglicht das Produkt den Netzwerkentwurf und dessen Dokumentation. So ist es möglich, automatisch Netzwerkgeräte in Schicht-2(Sicherungsschicht)-, Schicht-3(Vermittlungsschicht)- und Frame-Relay-Netzwerkumgebungen zu analysieren. Daneben eignet sich die Software zur Datenbankentwicklung sowie zur Erstellung verschiedener Software-Modelle mit Hilfe objektorientierter Notationen einschließlich aller acht UML-1.2-Diagrammtypen.

²Software Development Kit: Allgemein für „Programmierungsumgebung“. Programmentwicklungspaket, das Software-Bibliotheken, Werkzeuge und Dokumentationen zur Unterstützung von Anwendungsprogrammierern enthält.

MS Visio wurde in dieser Arbeit dazu verwendet, die Komponentendiagramme zu erstellen.

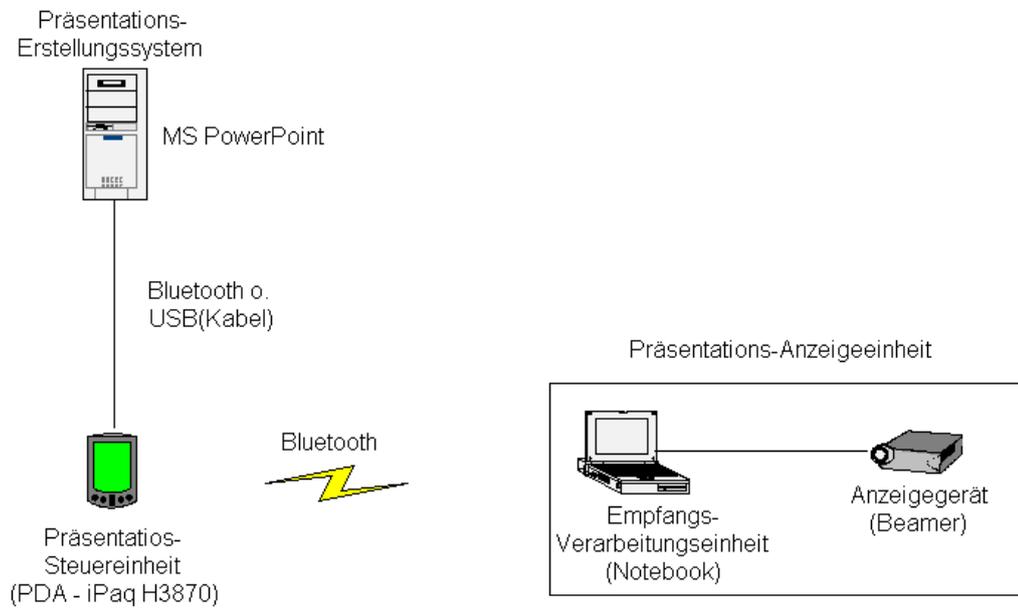


Abbildung 4.2: Gesamtüberblick der Systemarchitektur - 3.Version

Kapitel 5

Design und Realisierung

In diesem Kapitel wird zunächst die Systemarchitektur mit den einzelnen Systemkomponenten (PES, PSE und PAE) dargestellt. Desweiteren wird das Design und die Realisierung der einzelnen Systemkomponenten näher beschrieben. Zum Schluss des Kapitels gibt es eine Zusammenfassung, die die wichtigsten Inhalte des Kapitels in kurzer Form wiedergibt.

5.1 Systemarchitektur

Wie in der Abbildung: 5.1 zu sehen ist, besteht die Systemarchitektur aus drei Systemkomponenten, die alle ein eigenständiges Anwendungsprogramm darstellen sollen. Die drei Systemkomponenten heißen:

- PES - Präsentations-Erstellungssystem
- PSE - Präsentations-Steuereinheit
- PAE - Präsentations-Anzeigeeinheit

Unsere drei Systemkomponenten sind austauschbare Teile unseres Systems. Die ausgetauschten Systemkomponenten müssen lediglich unseren Schnittstellen-Spezifikationen genügen.

Desweiteren ist es möglich die Systemkomponenten auf verschiedener Hardware laufen zu lassen, z.B. ist die Systemkomponente PSE auf einem PDA, Webtablet, Handy, Desktop PC, usw. lauffähig.

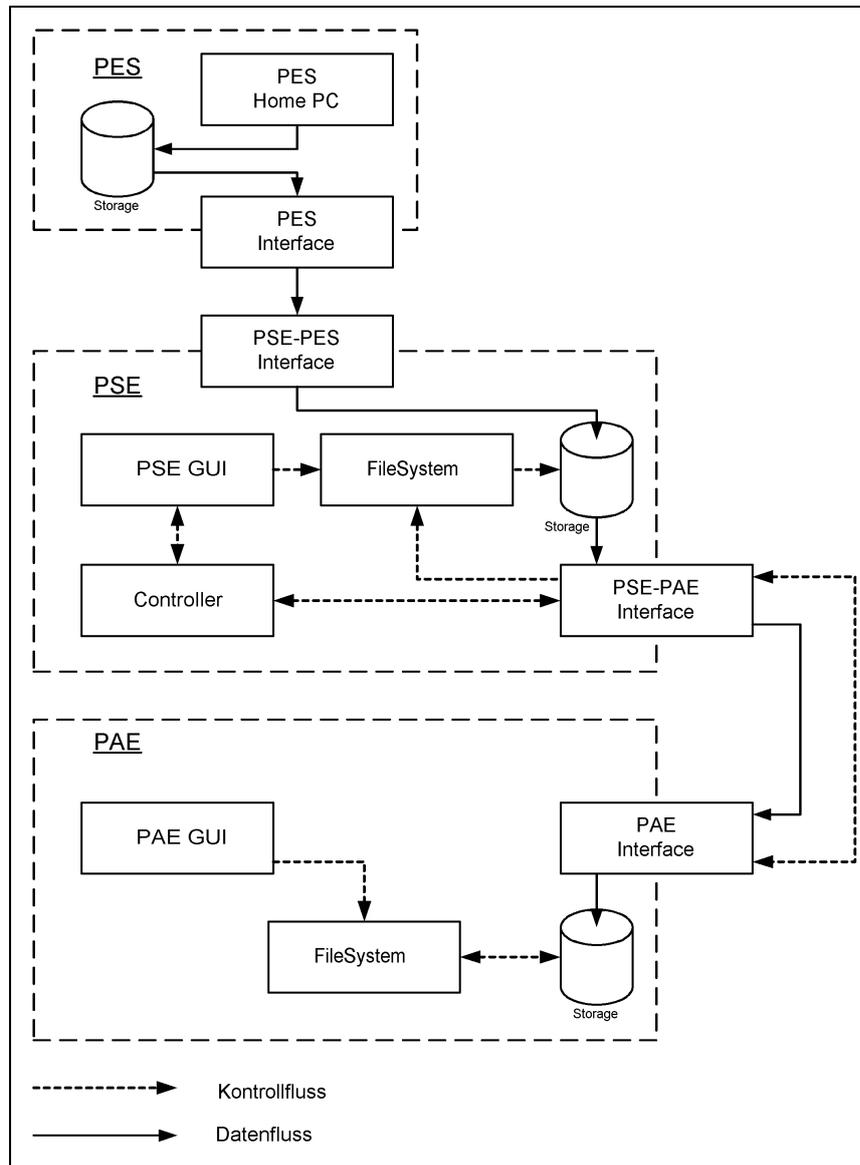


Abbildung 5.1: Komponentendiagramm

Die Systemkomponenten bestehen in unserem Fall aus geschachtelten Komponenten, und sind damit nicht hierarchisch oder netzwerkartig nebeneinander angeordnet, d.h. die Systemkomponenten werden aus einer Menge anderen Komponenten gebildet. Die Klassenbeziehungen müssen daher eine Aggregation oder Komposition darstellen und nicht etwa eine Generalisierung.

5.2 PES - Präsentations-Erstellungssystem

Dieses Kapitel wird in Software- und Hardware aufgeteilt.

- Software
 - PES - Home PC
 - PES - Interface
- Hardware
 - Desktop PC
 - Schnittstelle
(Bluetooth-USB-Adapter oder USB-Docking-Station)

Zunächst sprechen wir über die beiden Softwarekomponenten. Diese werden auf einem Rechnersystem: „Desktop PC“ laufen. Desweiteren muss das Rechnersystem eine Verbindung zum „PDA“ aufnehmen. Dafür wird eine weitere Hardware benötigt, die mit der Interfaceschnittstelle: „PES - Interface“ kommuniziert. Für die Kommunikation gibt es zwei Möglichkeiten. Die erste läuft über den Bluetooth-USB-Adapter und die zweite über die USB-Docking-Station.

Nähere Informationen zu den Hardwarekomponenten finden sich im Kapitel 4.1.

5.2.1 Software

PES - Home PC

Die erste Komponente: „PES - Home PC“ ist für die Erstellung und Speicherung der Präsentationsfolien zuständig. Für diese Komponente ist es nicht nötig, eine eigene Anwendung zu entwickeln, da bereits mehrere Anwendungen auf dem Markt vorhanden sind, die unsere Anforderungen erfüllen.

Vergleich von PES - Programmen

Die Produkte werden von uns nach verschiedenen Kriterien bewertet:

- Wirtschaftlichkeit (Was kostet das Produkt?)
- Konvertierungsmöglichkeit in andere Dateiformate
- Handhabbarkeit (vertraute Oberfläche und einfacher Umgang)

Mögliche Anwendungsprogramme sind:

1. MS¹ PowerPoint 97/2000/XP
2. StarOffice Impress 6.0
3. OpenOffice.org Impress 1.0

1. MS PowerPoint 97/2000/XP

- Vorteile:
 - Der Umgang mit MS PowerPoint ist für die meisten Leute vertraut, da sowohl im Geschäfts- als auch im Privatbereich überwiegend mit MS PowerPoint gearbeitet wird.
 - Konvertierung der Folien in verschiedene Dateiformate, wie z.B.: HTML, JPG, PPT, usw., ist bei MS PowerPoint 2000/XP möglich.
 - Bei MS PowerPoint XP ist es zusätzlich möglich, die Folien in XML zu konvertieren.
 - Lizenz für die deutsche Version von MS PowerPoint 2000 liegt bereits vor und braucht daher nicht mehr gekauft werden.
- Nachteile:
 - Bei MS PowerPoint 97/2000 ist keine Konvertierung in XML möglich.
 - Bei MS PowerPoint 97 ist keine Konvertierung in HTML möglich.

¹MS: Microsoft

2. StarOffice 6.0

- Vorteile:
 - Konvertierung der Folien in verschiedene Dateiformate ist bei StarOffice möglich, wie z.B. in HTML, JPG, PPT usw. Zusätzlich ist auch die Konvertierung in XML vorhanden.
- Nachteile:
 - StarOffice ist seit Anfang 2002 kostenpflichtig und es liegt uns keine Lizenz vor.
 - Da StarOffice auf dem Markt nicht so weit verbreitet ist, ist eine gewisse Einarbeitungszeit für die meisten Benutzer nötig.

3. OpenOffice 1.0

- Vorteile:
 - OpenOffice ist kostenlos, daher wird keine Lizenz benötigt.
 - Konvertierung der Folien in verschiedene Dateiformate, wie bei StarOffice, möglich.
- Nachteile:
 - Da auch OpenOffice auf dem Markt nicht so weit verbreitet ist, ist eine gewisse Einarbeitungszeit für die meisten Benutzer nötig.

Auswahl des Dateiformates

Aus folgenden Gründen haben wir uns für das Dateiformat JPG entschieden:
Im Zusammenhang mit dem Präsentationserstellungsprogramm hat das Dateiformat JPG einige wichtige Vorteile:

- Es ist möglich auch mit älteren Programmen, wie z.B. MS PowerPoint 97, die Präsentation als JPG zu speichern (die Konvertierung in HTML ist hiermit z.B. nicht möglich).
- Bei der Konvertierung des Foliensatzes wird jede einzelne Folie als ein JPG abgespeichert, dadurch kann die Präsentation schon nach der Übertragung, der ersten Folie, zur PAE gestartet werden.
- Desweiteren werden die Folien unter einem fest definierten Dateinamen gespeichert (z.B. bei MS PowerPoint 2000 als Folie1.jpg). Somit sind die Folien auch automatisch in der richtigen Reihenfolge.

- Ausserdem kann zur Erstellung der Präsentation auch jedes andere Programm benutzt werden, das die Speicherung der Folien als JPG unterstützt. So ist es z.B. auch möglich mit MS Visio erstellte Zeichnungen als JPG abzuspeichern und diese als Präsentation vorzuführen. Auch eine Diashow von mehreren Bildern (Fotos) ist mit JPG denkbar.

Ein großer Nachteil von JPG ist allerdings, das keinerlei Animationen in die Präsentation eingebaut werden können, bzw., wenn die Präsentation mit Animationen erstellt wurde, können dies nicht dargestellt werden.

Warum nicht PPT?

PPT ist das Dateiformat, das MS Powerpoint erstellt. Hierbei wird die Präsentation als eine Datei abgespeichert. Das heißt, das zuerst die ganze Datei zur PAE übertragen werden muss, bevor mit der Präsentation begonnen werden kann und das würde einen zu großen Zeitaufwand bedeuten. Desweiteren ist die Steuerung der Präsentation sehr schwierig, da die Syntax der PPT-Datei, von Microsoft, nicht öffentlich gemacht wurde.

Warum nicht HTML² oder XML³?

Ältere Programme können nicht in diese Formate konvertieren, z.B. MS Powerpoint 97. Bei HTML sowie XML werden die Präsentationsfolien nicht als einzelne Folien sondern in mehrere Dateien unterschiedlichen Formats abgespeichert. Auch die Reihenfolge der einzelnen Folien ist anhand der Dateinamen nur schwer zu erkennen. Dadurch müsste man erst den gesamten Foliensatz übertragen, bevor mit der Präsentation begonnen werden kann.

Auswahl des PES - Programms

Wir benötigen ein Anwendungsprogramm zur Erstellung der Präsentation und zur Konvertierung der einzelnen Folien in das Format JPG. Hierfür benutzen wir aus folgenden Gründen MS PowerPoint 2000:

- Das Erstellen der MS PowerPoint 2000 Folien und das Konvertieren in JPG stellt durch die vertraute Oberfläche und dem leichten Umgang kein Problem dar. Aus diesem Grund ist keine Einarbeitungszeit nötig.

Für die zukünftige Änderbarkeit ist die Konvertierung der Folien, wie schon beschrieben, in andere Dateiformate möglich. Sollte jedoch eine Konvertierung in XML erwünscht werden, muß entweder auf MS Powerpoint XP oder auf OpenOffice 1.0 umgestiegen werden. Wenn eine Lizenz für MS Powerpoint vorliegt, sollte man sich hierfür entscheiden. Ansonsten ist ein Umstieg auf das kostenlose OpenOffice 1.0 sinnvoll, man muß nur eine gewisse Einarbeitungszeit mit einberechnen.

²HyperText Markup Language - Darstellungs-Standard für Dokumente im Internet

³Extensible Markup Language - Metasprache, mit der sich neue Darstellungssprachen definieren lassen

PES - Interface

Die zweite Komponente: „PES - Interface“ ist für die Übertragung der Präsentationsfolien vom PES zur PSE zuständig.

Nachdem der Foliensatz, mit MS Powerpoint 2000, erstellt und in einen Ordner (der Ordner hat den selben Namen wie die enthaltene Präsentation) auf dem PES gespeichert wurde, muß dieser Foliensatz anschließend zur PSE übertragen werden.

Es wird also ein Programm benötigt, welches in der Lage ist, den Foliensatz-Ordner mit den enthaltenen Folien zur PSE zu übertragen und dort in einen definierten Unterordner des Hauptprogramms abzuspeichern.

Wie schon im Kapitel 5.2.1 beschrieben, werden die einzelnen Folien als JPG-Dateiformat abgespeichert. Die Dateinamen der Folien sind definiert: folie1.jpg bis folieN.jpg

Zur Übertragung der Folien benutzen wir die Software MS ActiveSync, die beim Compaq iPAQ H3870 im Lieferumfang mit dabei ist, weil dieses Produkt unseren Anforderungen genügt und von Microsoft gepflegt wird.

5.2.2 Hardware

Um die Präsentation zu erstellen und anschließend zum PDA übertragen zu können werden noch ein Desktop PC und eine physikalische Verbindung vom Desktop PC zum PDA benötigt.

Desktop PC

Beim Desktop PC haben wir uns für den, im Kapitel 4.1 Home PC beschriebenen, PC entschieden, da er in unserem Labor zur Verfügung steht und unseren Ansprüchen genügt. Auf diesem PC können wir MS PowerPoint 2000 benutzen und es ist eine USB-Schnittstelle für die Docking-Station oder für den USB-Bluetooth-Adapter vorhanden.

Schnittstelle

Hier stehen zwei Möglichkeiten zur Verfügung:

- USB-Docking-Station
- USB-Bluetooth-Adapter

Die einfachere Lösung ist hier die USB-Docking-Station (auch Cradle genannt), da diese schon im Lieferumfang des PDAs vorhanden ist und ohne Probleme über ein Kabel am USB-Anschluss des PCs angeschlossen werden kann. Nun braucht der PDA nur noch ins Cradle gesteckt zu werden, um eine physikalische Verbindung zum PDA herzustellen.

Eine andere Möglichkeit ist der USB-Bluetooth-Adapter. Dieser wird ebenfalls an die USB-Schnittstelle des PCs angeschlossen. Desweiteren müssen noch einige Einstellungen für die COM-Ports vorgenommen werden, um eine Verbindung zwischen dem PC und dem PDA herstellen zu können. Eine genaue Erklärung und einen Erfahrungsbericht mit dieser Technologie finden sie im Anhang.

5.3 PSE - Präsentations-Steuereinheit

Hier wird die zweite Systemkomponente beschrieben, die auch wieder in Software- und Hardware aufgeteilt wird.

- Software
 - Sequenzdiagramme
 - Klassendiagramm
 - Systemkomponenten
- Hardware
 - PDA

In diesem Kapitel werden wir näher auf die fünf Softwarekomponenten, wie in Abbildung 5.8 zu sehen, eingehen. Diese werden auf einem mobilen Rechnersystem laufen. In unserem Fall wird es der PDA Compaq IPaq H3870 sein, den wir im Hardware-Abschnitt näher beschreiben werden. Desweiteren gehen wir hier auf zwei Schnittstellen ein, die erste kommuniziert mit dem PES und die zweite mit dem PAE.

Nähere Informationen zu den Hardwarekomponenten finden Sie im Kapitel 4.1.

Um alle nötigen Klassen, Methoden und Attribute finden zu können, wurden Sequenzdiagramme erstellt, die den zeitlichen Ablauf mit ihren Interaktionen zwischen den Klassen darstellen sollen. Um den zeitlichen Ablauf der Sequenzdiagramme besser verstehen zu können, ist jedem Diagramm ein Dialogfenster zugeordnet. Das Dialogfenster zeigt die Ansicht des Benutzers.

5.3.1 Software

Sequenzdiagramme

In diesen Diagramm spielt der zeitliche Ablauf der Interaktionen, sowie das dynamische Verhalten der Klassen zueinander eine große Rolle, um Klassen, Methoden und deren Attribute ausfindig machen zu können. Daher wird das Sequenzdiagramm mit zusätzlichen Kommentaren erklärt, um diese besser verstehen zu können. Desweiteren werden Dialogfenster aus der Anwendersicht den Sequenzdiagrammen zugeordnet. Daher sind, wenn möglich, die Abbildungen mit Sequenzdiagramm und Dialogfenster zusammen abgebildet.

Die folgenden Sequenzdiagramme werden nun aufgelistet:

1. Sequenzdiagramm Nr.1: Folien- und Bluetoothgeräteauswahl
2. Sequenzdiagramm Nr.2: Verbindungsaufbau und Streaming der Folien
3. Sequenzdiagramm Nr.3: Folienanzeige auf dem PAE
4. Sequenzdiagramm Nr.4: Diashow Auswahl der Profil-Einstellungen
5. Sequenzdiagramm Nr.5: Diashow Ablauf

Nr.1: Folien- und Bluetoothgeräteauswahl

Zur Veranschaulichung des Textes, siehe Abbildung 5.2.

Nach dem Start des Programms müssen erst einmal der Foliensatz und das Bluetoothgerät ausgewählt werden. Hier wird davon ausgegangen, das sich im Umkreis von 10m weitere Bluetoothgeräte befinden und der Benutzer keinen Offlinebetrieb, ohne Bluetoothgerät wünscht.

Alle Methoden der *CBtStack* Klasse werden im Labor im Unterabschnitt 4.2.3 Bibliotheken beschrieben. Als erstes initialisiert die Methode 1:*Connect()* eine Verbindung zum Bluetooth Stack. Diese Methode gehört zu den *BTAccess* Klassen. Danach wird mit der Methode 2: *listenSlideFolders()* eine Liste der Foliensätze-Ordner angezeigt. Diese befinden sich im folgenden-Pfad: „*/easypresenter/folien*“. Wie auf der Abbildung 5.2 sichtbar, könnte z.B. in der Listbox: „*Foliensatz auswählen*“ oben, folgendes stehen:

1. Vorlesung KI
2. Vorlesung Robbi

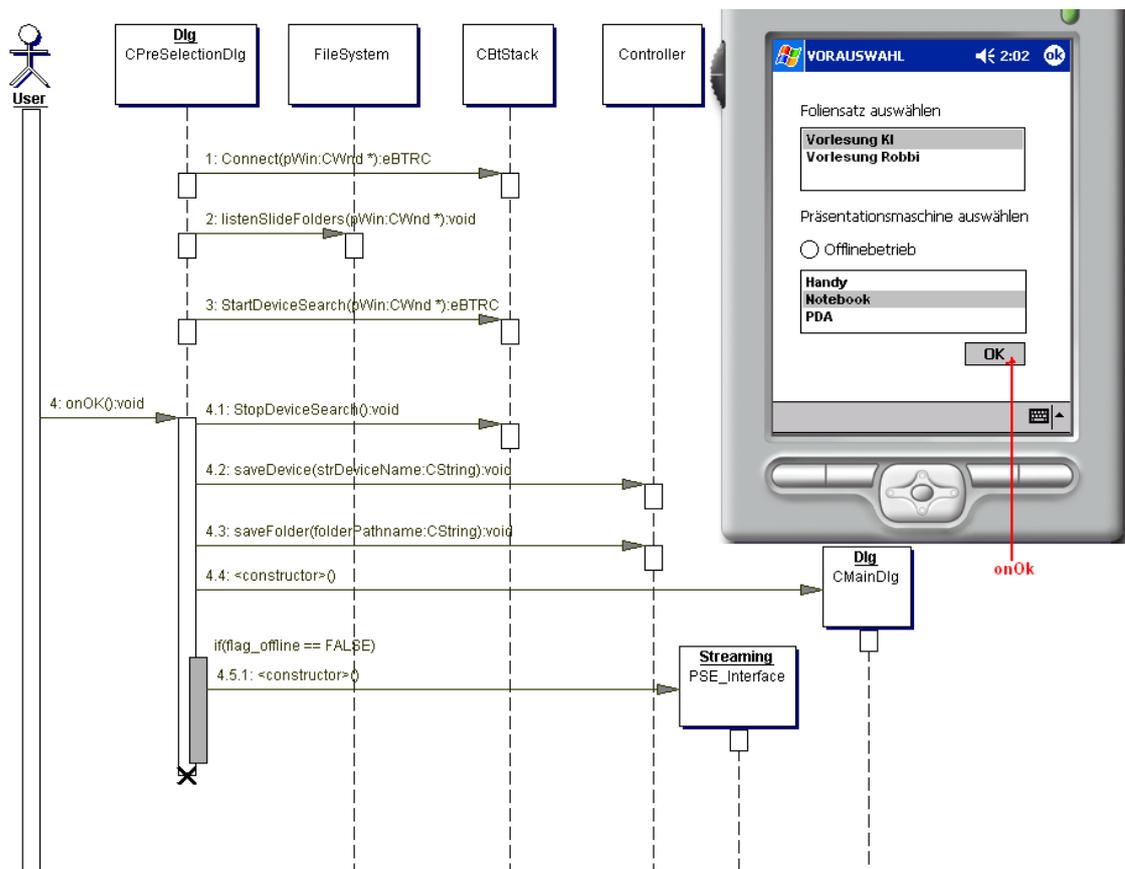


Abbildung 5.2: Folien- und Bluetoothgeräteauswahl

Der Anwender wählt eins der beiden Foliensätze durch Selektieren aus.

Gleich danach wird die Methode 3: *StartDeviceSearch()* der Klasse *CBtStack* aufgerufen. Diese Methode sorgt dafür, das im Umkreis von ca. 10 m nach anderen Bluetoothgeräten gesucht wird. Die Ergebnisse werden in der unteren „Listbox“ des Vorauswahl-Dialogfensters aufgelistet.

Nachdem der Benutzer einen Foliensatz und ein Bluetoothgerät ausgewählt hat, betätigt dieser den Button „OK“, dadurch wird die Methode 4: *onOk()* der Klasse *CPreSelectionDlg* ausgeführt.

Die Methode erledigt folgende Aufgaben:

1. Sie unterbricht die Suche nach anderen Bluetoothgeräten, durch den Methodenaufruf 4.1: *StopDeviceSearch()*
2. Der Gerätenamen des ausgewählten Bluetoothgerätes wird mit dem Methodenaufruf 4.2: *saveDevice()* gespeichert.

3. Der ausgewählte Foliensatz-Ordner wird mit kompletter Pfadangabe: “/easypresen-ter/folien/vorlesungki“ mit dem Methoden-Aufruf 4.3: *saveFolder()* gespeichert.
4. Mit dem Konstruktor-Aufruf 4.4 <constructor>(), wird ein Objekt *Dlg* der Klasse *CMainDlg* erzeugt.
5. Wenn das Flag für den Offlinebetrieb nicht gesetzt ist, wird das Objekt *Streaming* der Klasse: *CPSE_interface* mit dem Konstruktor-Aufruf 4.5.1: <constructor>() initiiert.

Zum Schluß wird die Instanz zerstört. Damit wird das Fenster für die Vorauswahl wieder geschlossen und das Hauptfenster, welches mit dem Konstruktor 4.4 <constructor>() erzeugt wurde, wird sichtbar.

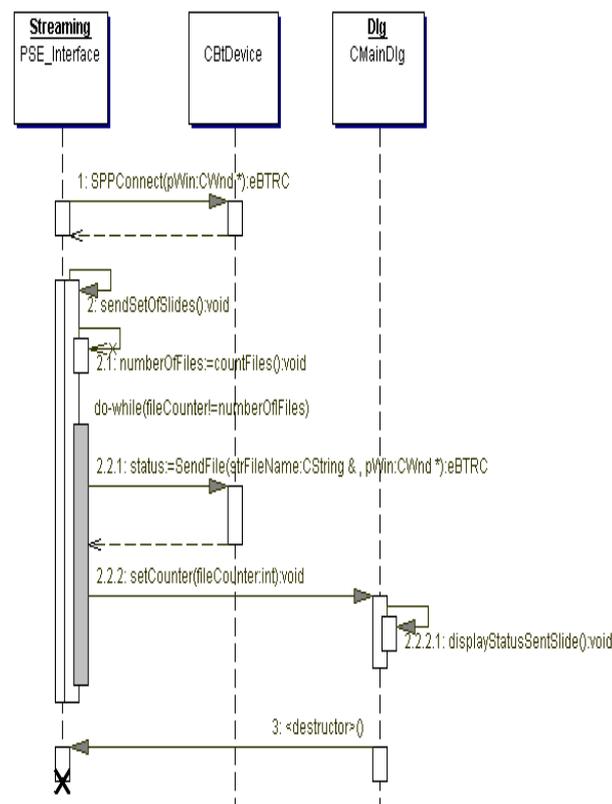


Abbildung 5.3: Verbindungsaufbau zum Bluetoothgerät und Streaming der Dateien

Nr.2: Verbindungsaufbau und Streaming

Zur Veranschaulichung des Textes, siehe Abbildung 5.3.

Nachdem alle Einstellungen vom Anwender im Vorauswahlfenster mit dem „Ok“-Button bestätigt und übernommen wurden, wird zusätzlich in der *onOk()* Methode, der Klasse:

CPreSelectionDlg, nach Abfrage des Offline-Flags, eine Instanz, namens *Streaming*, von der Klasse *CPSE_interface* erzeugt.

Nachdem die Instanz erzeugt wurde, wird als erstes die Methode 2: *SPPConnect()* aufgerufen. Hierbei wird ein Verbindung über Bluetooth per *Seriellen-Port-Profil(SPP)* zum anderen Gerät, hier der PAE, hergestellt.

Als nächstes wird die Methode 2: *sendSetOfSlides()* aufgerufen. Diese ist zuständig für die Übertragung der einzelnen Folien zum PAE und das Setzen des *fileCounters* für die schon übertragenen Folien.

Die While-Schleife wird solange ausgeführt, bis alle Folien übertragen wurden, also bis der *fileCounter() == countFiles()* True ist.

1. Die erste Methode in der While-Schleife ist für die Übertragung der einzelnen Folien zuständig 2.2.1 *SendFile()*. Sie überträgt per FTP⁴ die einzelnen Folien-Dateien zum PAE.
2. Die zweite Methode 2.2.2 *SetCounter()* erhöht jedesmal, wenn eine Folie übertragen wurde, die *fileCounter* Variable. Desweiteren kann man mit der Abfrage dieser Variable vor Aufruf einer Folie auf dem PAE abfragen, ob diese schon übertragen wurde. Desweiteren wird in der Methode 2.2.2.1 *displayStatusSentSlide()* aufgerufen, um den neuen Status auf dem Display des PDA's anzeigen zu können. Somit ist es möglich, den Anwender über den Status der übertragenen Folien zu informieren.
3. Die dritte Methode 2.2.2.1 *displayStatusSentSlide()*

Nr.3: Folieanzeige auf der PAE

Zur Veranschaulichung des Textes, siehe Abbildung 5.4.

Die Vorbedingungen:

1. Foliensatz und Bluetoothgerät für die Präsentationsanzeige wurde ausgewählt.
2. Option für Anzeigebestätigung ist aktiv. Das heißt, jede Folie auf dem PDA muß erst mit einer Anzeigebestätigungstaste vom Anwender bestätigt werden, damit die Folie auf dem PAE angezeigt werden kann.

Nachdem der Anwender den „forward“ Button betätigt, wird die Methode 1:*slideForward()*, der *CMainDlg* Klasse, aufgerufen. In dieser Methode wird als erstes die Methode 1.1: *showNextSlide()* aufgerufen. In dieser Methode wiederum wird der aktuelle

⁴FTP ist eine Abkürzung für File Transfer Protokoll

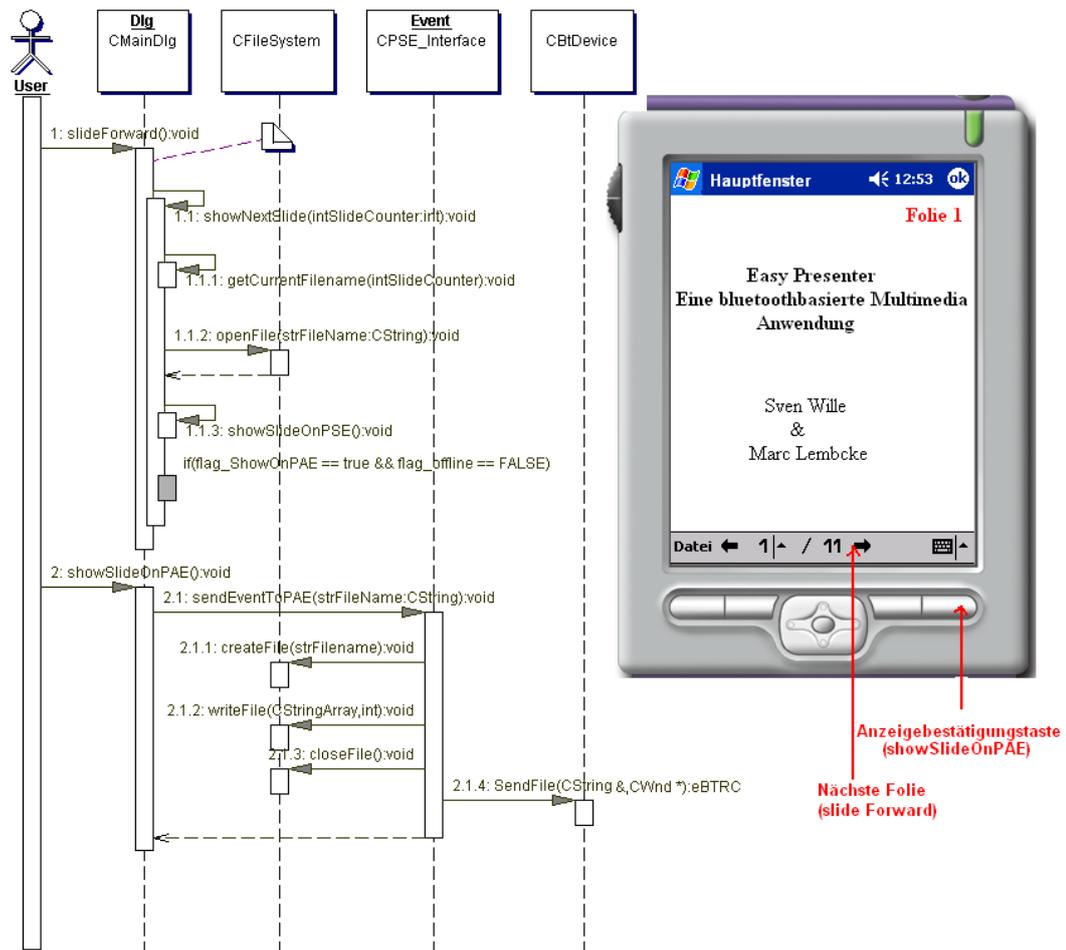


Abbildung 5.4: Folienanzeige auf der PAE

Folien-Dateiname mit der Methode 1.1.1: *getCurrentFilename()* bestimmt. Hierfür wird eine Variable *intSlideCounter* benötigt. Diese Variable zeigt die aktuelle Foliennummer an, die als nächstes auf dem PSE angezeigt werden soll. Wenn also z.B. der Anwender den „forward“ Button betätigt, wird die Variable um eins inkrementiert oder bei Betätigung des „backward“ Buttons um eins dekrementiert. Die Dateinamen für die Folien sind aufwärtssteigend durchnummeriert. Die Syntax der Dateinamen sieht folgendermaßen aus: *Folie001*, *Folie002* bis *Foliexyz*. Durch die Verknüpfung des Wortes „Folie“+“*intSlideCounter*“ lässt sich der aktuelle Folien-Dateiname in der Methode *getCurrentFilename* feststellen. Nachdem der Dateiname in dem String *strFileName* zugewiesen wurde, kann mit der Methode 1.1.2: *openFile()*, diese Folie geöffnet werden. Mit der Methode 1.1.3: *showSlideOnPSE()* wird die Folie auf dem Display des PSE angezeigt.

Da das Anzeigebestätigungs-Flag *flag_ShowOnPAE* nicht gesetzt ist, bzw. jede Folie vom Anwender bestätigt werden muß, wird die Folie auf dem PSE angezeigt, aber auf dem PAE erst, nachdem der Anwender die Anzeigebestätigungs-Taste auf dem PDA betätigt

hat. Bei Betätigung dieser Anzeigebestätigungstaste wird die Methode 2: *showSlideOnPAE()* aufgerufen und mit dieser wird die Folie auf dem PAE angezeigt. In dieser Methode wird eine Text-Datei, die den Dateinamen der anzuzeigenden Folie beinhaltet, zur PAE übermittelt. Da die BTAccess Klassen noch keine Methode zur Verfügung stellen, um ein Text-String zu senden, gibt es nur zwei Möglichkeiten:

1. Öffnen einer zweiten TCP/IP Verbindung über *Winsock*.
2. Benutzung der Methode *SendFile()* aus der *CBTDevice* Klasse, da Verbindung über SPP schon vorhanden ist.

Da es ein größerer Aufwand ist, erst eine zweite Verbindung über die *Winsock* Klassen zu erstellen, wurde hier die zweite Möglichkeit gewählt. Hierbei wird die Information über den Dateinamen der Folie, die als nächstes angezeigt werden soll, in eine Datei gespeichert. Die Datei hat folgende Extension: *.txt*. Die Erzeugung der Datei und die Speicherung der Information (Foliennamen) in diese Datei, passiert mit den Methoden 2.1.1: *createFile()* und 2.1.2: *writeFile()*. Nachdem die Information in die Datei gespeichert wurde, muß sie mit 2.1.3: *closeFile()* wieder geschlossen werden. Zum Schluß kann die Datei mit der Methode 2.1.4: *SendFile()* zum PAE übertragen werden.

Zum Abschluß ist noch zu sagen, das es noch weitere Varianten der Anzeige von Folien auf dem PAE gibt. Da diese sich aber sehr gleichen, wurde nicht extra für jede Variante ein Sequenzdiagramm erstellt.

1. Benutzer betätigt den „*slideBackward*“ Button. Hier bleibt alles gleich, bis auf die Variable *intSlideCounter*, die um eins dekrementiert wird. Die Methode heißt: *slidebackward()*
2. Benutzer wählt „selektiv“ eine bestimmte Folie aus. Hierbei wird der Variable *intSlideCounter*, die Foliennummer, die vom Anwender selektiv ausgewählt wurde, zugewiesen. Die Methode heißt: *slideSelective(intSlideNumber)* und die ausgewählte Foliennummer wird mit dem Parameter: *intSlideCounter* übergeben.

Nr.4: Diashow Auswahl

Zur Veranschaulichung des Textes, siehe Abbildung 5.5.

Die Vorbedingungen:

1. Foliensatz und Bluetoothgerät für die Präsentationsanzeige wurde ausgewählt.
2. Im Hauptfenster wurde vom Anwender unter Menüpunkt: „Datei“ die Diashow ausgewählt.

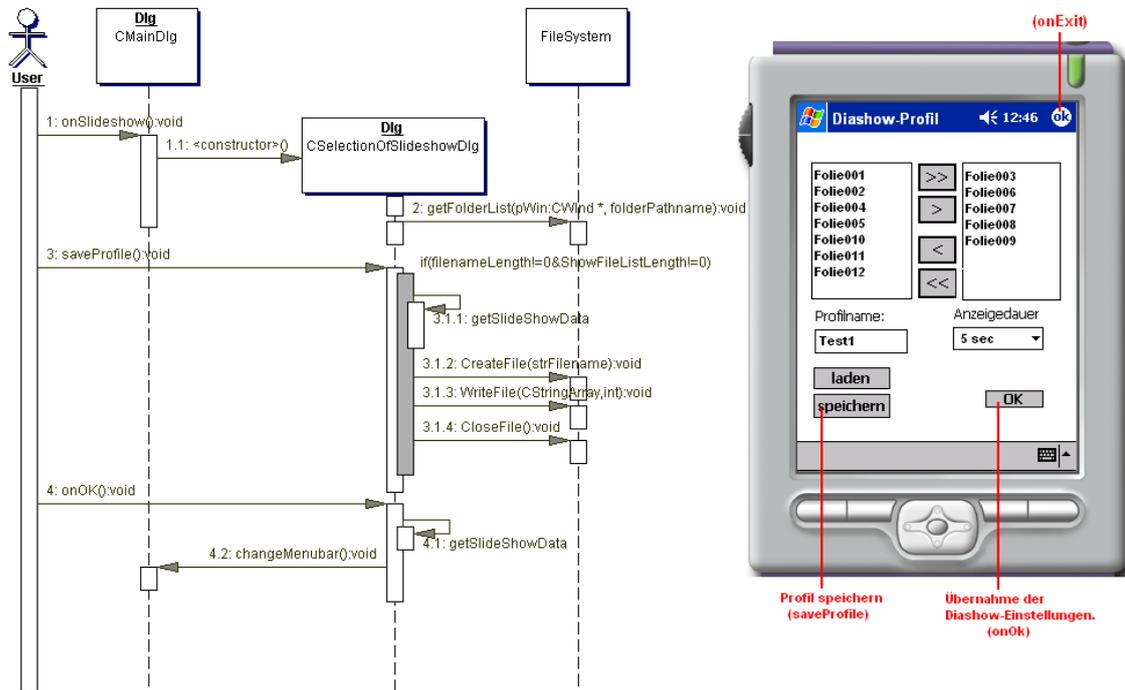


Abbildung 5.5: Diashow Auswahl

Durch die Auswahl der Diashow wird ein neues Dialogfenster für die Einstellungen der Diashow geöffnet. Dies passiert in der Methode: 1: *onSlideshow()*, der Klasse *CMainDlg*. In dieser Methode wird der Konstruktor für die Klasse *CSelectionOfSlideshowDlg* aufgerufen. Dieser erzeugt eine Instanz namens *Dlg* von der genannten Klasse.

In der Instanz *Dlg* wird die Methode 2: *getFolderList()* ausgeführt, um die vorhandenen Folien im „Diashow Auswahlfenster“ anzeigen zu lassen. Der komplette Foliensatz-Ordner mit allen Folien wird auf der linken Seite des Dialogfensters angezeigt. Der Anwender kann bestimmte Folien in die rechte Anzeigebox verschieben, dies funktioniert mit den zwei Rechts- und Linksschiebebutton. Nur die Folien in der rechten Anzeigebox werden für die Diashow in der zu sehenden Reihenfolge verwendet. Nachdem der Anwender alle Folien ausgewählt und die Anzeigedauer eingestellt hat, kann er einen Profilnamen in die „Editbox“ schreiben und die gemachten Einstellungen mit dem *Speichern* Button abspeichern. Beim Abspeichern des Profils wird die Methode 3: *saveProfile()* ausgeführt. Bevor gespeichert werden kann, muß der Button „speichern“ erst aktiv werden. Dies passiert, wenn ein Dateiname für das Profil eingegeben wurde und in der Anzeigeliste (rechte Box im Dialogfenster) Folien für die Diashow ausgewählt wurden. Wenn die beiden Bedingungen erfüllt sind und der „speicher“ Button betätigt wurde, holt die Methode 3.1.1: *getSlideShowData()* sich die vom Anwender eingestellten Daten.

Mit 3.1.2 *CreateFile()* wird eine Datei mit den angegebenen Profilnamen erzeugt und geöffnet. Mit 3.1.3 *WriteFile()* werden die gemachten Daten als Text hintereinander, durch

Kommas, getrennt in die Profil-Datei gespeichert. Nachdem alle Daten in die Datei geschrieben wurden, muß diese mit 3.1.4 *CloseFile()* wieder geschlossen werden.

Mit der Betätigung des „OK“ Buttons werden alle gemachten Einstellungen für die jetzige Diashow, mit 4.1: *getSlideShowData()*, übernommen. Das Hauptfenster im Hintergrund wird mit der Diashow-Button Oberfläche geladen. Das passiert mit 4.2: *changeMenubar()*. Durch Beenden des Dialogfensters, der „Diashow-Auswahl“, wird das Hauptfenster mit der geänderten Menüleiste für die Diashow sichtbar.

Die Variante „Laden eines Profils“ wurde hier nicht gezeigt. Diese ist aber sehr ähnlich mit der des „Speichern eines Profils“. Die Methoden 3.1.2 *CreateFile()* und 3.1.3 *WriteFile()* müssen lediglich nur mit den Methoden 3.1.1 *OpenFile()* und 3.1.2 *ReadFile()* ausgetauscht werden. Die Methode 3.1.1: *getSlideShowData()* wird nicht gebraucht. Stattdessen muss am Ende nur eine Methode 3.1.3 *setSlideShowData()* benutzt werden. Diese Methode sorgt dafür, das im Dialogfenster alle Einstellungen für den Anwender sichtbar gemacht werden.

Desweiteren muß auch gewährleistet werden, das der Anwender, ohne Übernahme irgendwelcher Einstellungen, zurück ins Hauptfenster kommt, z.B. wenn der Anwender sich vertan hat und keine Diashow wünscht. Dies passiert durch Drücken des „Exit-Icons“ oben rechts in der Ecke mit Schriftzug: „ok“. Hierdurch wird die Methode *onExit()* aufgerufen. Das Programm soll nicht beendet werden, nur das aktive Fenster wird geschlossen und das Hauptfenster wieder sichtbar.

Nr.5: Diashow Hauptfenster

Zur Veranschaulichung des Textes, siehe Abbildung 5.6.

Die Vorbedingungen:

1. Foliensatz und Bluetoothgerät für die Präsentationsanzeige wurde ausgewählt. „Offlinebetrieb“ wurde nicht ausgewählt.
2. Das Flag: *flag_ShowOnPAE* ist bei der Diashow auf „True“ gesetzt, da der Anwender die Automatik-Option, zum Anzeigen auf dem PAE, ausgewählt hat.
3. Im „Diashow-Auswahlfenster“ wurden alle Einstellungen gemacht oder ein Profil mit allen Einstellungen wurde geladen.
4. Das „Diashow-Auswahlfenster“ wurde beendet und das Hauptfenster aus dem Hintergrund ist wieder aktiv.
5. Das „Diashow-Auswahlfenster“ wurde in der Menübar im Hauptfenster geändert. Die Menübar „Datei“ bleibt erhalten, der Rest wurde gegen einen „Play/Pause“ und einen „Stop“ Button, im letzten Sequenzdiagramm Nr.4, getauscht.

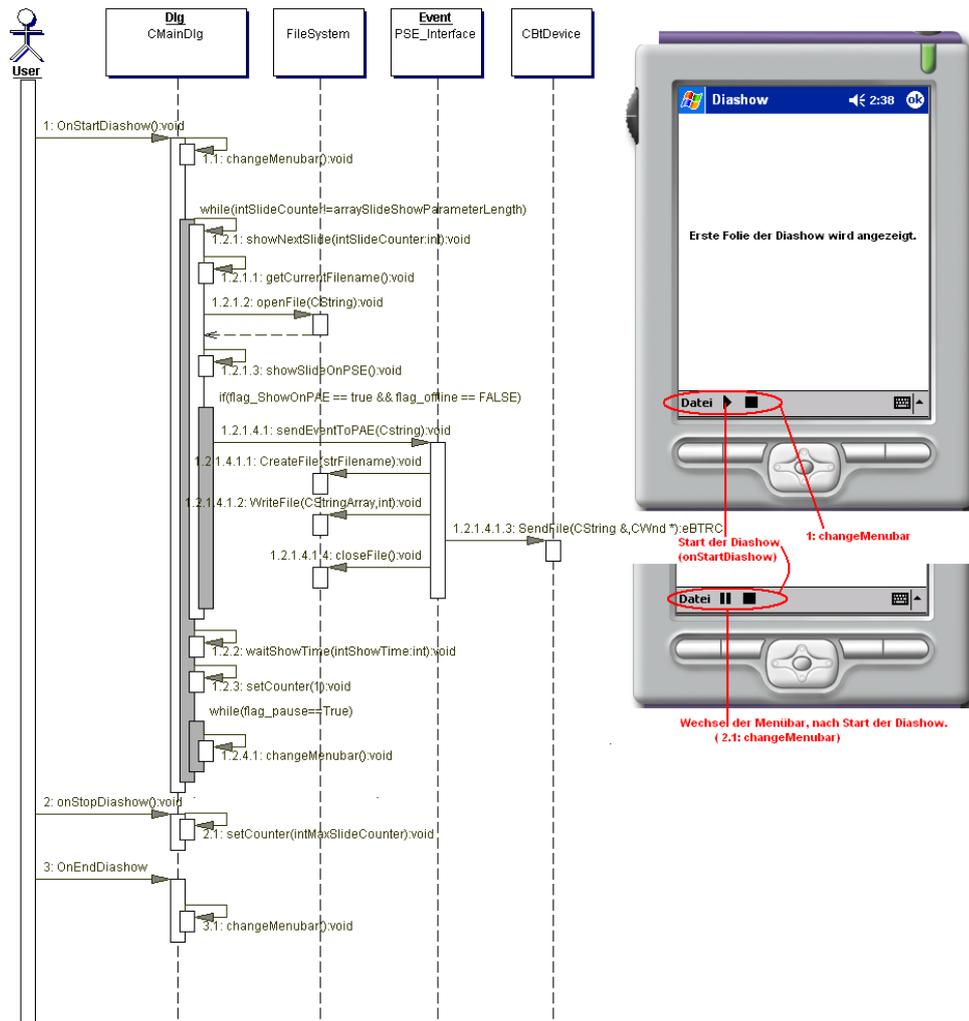


Abbildung 5.6: Diashow Hauptfenster

6. Eine Verbindung zum PAE über SPP besteht bereits. Daher wurde die Methode *changeMenubar()* weggelassen.

Die Methode 1: *onStartDiashow()*, wird durch Betätigen des „Play“ Buttons vom Anwenders ausgelöst und dadurch die Diashow gestartet.

Folgende Punkte werden in dieser Methode erledigt:

1. Nachdem die Diashow gestartet wurde, ändert sich der „Play“ Button zum „Pause“ Button. Hierfür ist 1.1: *changeMenubar()* zuständig. Bei erneutem Drücken des Buttons kann die Diashow unterbrochen werden. Dazu wird eine Methode: *onPause()* ausgeführt, hier im Sequenzdiagramm: „Diashow Hauptfenster“ nicht mit

aufgeführt. Diese setzt eine Flag-Variabel: *flag_Pause*. Diese Variabel wird später in einer 1.2.4 While-Schleife abgefragt. Solange dieser Flag auf True gesetzt ist, verweilt die Methode in dieser While-Schleife. Nach erneutem Betätigen des „Play/Pause“ Buttons wird die Diashow fortgesetzt.

2. In einer While Schleife wird eine nach der anderen Folien, nach Reihenfolge, auf dem Display des PSE geöffnet und als „Event-Datei“ zum Anzeigegerät, per FTP über SPP, gesendet. Auf der Seite des PAE, wird die Datei geöffnet und die Information aus der Datei gelesen. Mit der gelesenen Information, wird die aktuelle Folie geöffnet und auf dem Anzeigegerät, hier ein Beamer, ausgegeben.
3. Nachdem die „Event-Datei“ gesendet wurde, wird eine bestimmte Zeit gewartet. Die Wartezeit entspricht der Anzeigedauer, die im Profil für die „Diashow-Auswahl“ eingestellt wurde. Dies passiert in der Methode 1.2.2:*waitShowTime()*

Zu Punkt zwei:

Die Folien-Namen sind in einem Array abgelegt. Eine Counter-Variabel *intSlideCounter* zeigt auf den Platz des Arrays für die nächste anzuzeigende Folie. Dieses geschieht in der Methode 1.2.1: *showNextSlide()* In dieser Methode wird als erstes der Dateiname der aktuellen Folie aus dem Array geholt, mit 1.2.1.1 *getCurrentFilename()*. Danach wird die Datei mit 1.2.1.2 *openFile()* geöffnet und auf dem Display des PDA's angezeigt. Alles weitere läuft wie im Sequenzdiagramm Nr.3: Folienanzeige auf der PAE ab. Siehe hierzu Abbildung 5.4. Ein Unterschied gibt es doch. Bei 1.2.1.4.1: *sendEventToPAE()* wird der „Event-String“ in einer „Event-Datei“ ohne Anzeigebestätigung des Anwenders zum PAE übertragen. Nachdem die Folie auf dem PAE angezeigt wurde, wird durch die Methode 1.2.2: *waitShowTime()*, nach Vorgabe der Anzeigedauer, gewartet. Zum Schluß der While-Schleife wird das „Pause“ Flag in einer weiteren While-Schleife abgefragt. Hat der Anwender den „Pause“ Button gedrückt, wird hier solange gewartet, bis der „Play“ Button erneut betätigt wird. Beim Betätigen des „Play“ und des „Pause“ Buttons wird jedes mal die Menübar neu mit der Methode „changeMenubar()“ geladen.

Was passiert, wenn der Anwender den „Stop“ Button betätigt?

Wird der Button „Stop“ während der Diashow vom Anwender bedient, sorgt die Methode 2: *onStopDiashow()* dafür, das die Counter-Variabel : *intSlideCounter* auf das Ende der letzten Foliennummer gesetzt wird, also auf *arraySlideShowParameterLength*. Hierdurch wird die Methode 1: *onStartDiashow()* beendet. Mit *changeMenubar()* wird die Menübar mit dem „Play“ Button wieder angezeigt. Durch erneutes Betätigen des „Play“ Buttons wird die Präsentation von vorne gestartet.

Wie beendet man die Diashow?

Durch Drücken auf das „ok“-Icon, in der rechten oberen Ecke des Dialogfensters, wird die Methode 3: *onEndDiashow* ausgeführt. Diese Methode sorgt, wie die Methode 2: *onStopDiashow()* dafür, das die Methode 1: *onStartDiashow()* beendet wird. Ausserdem wird die

Menübar auf die Grundeinstellung des Hauptfensters zurück verändert. Das „Diashow-Hauptfenster“ ist also nicht mehr mit seinen „Play/Pause“ und „Stop“ Button sichtbar.

Klassendiagramm

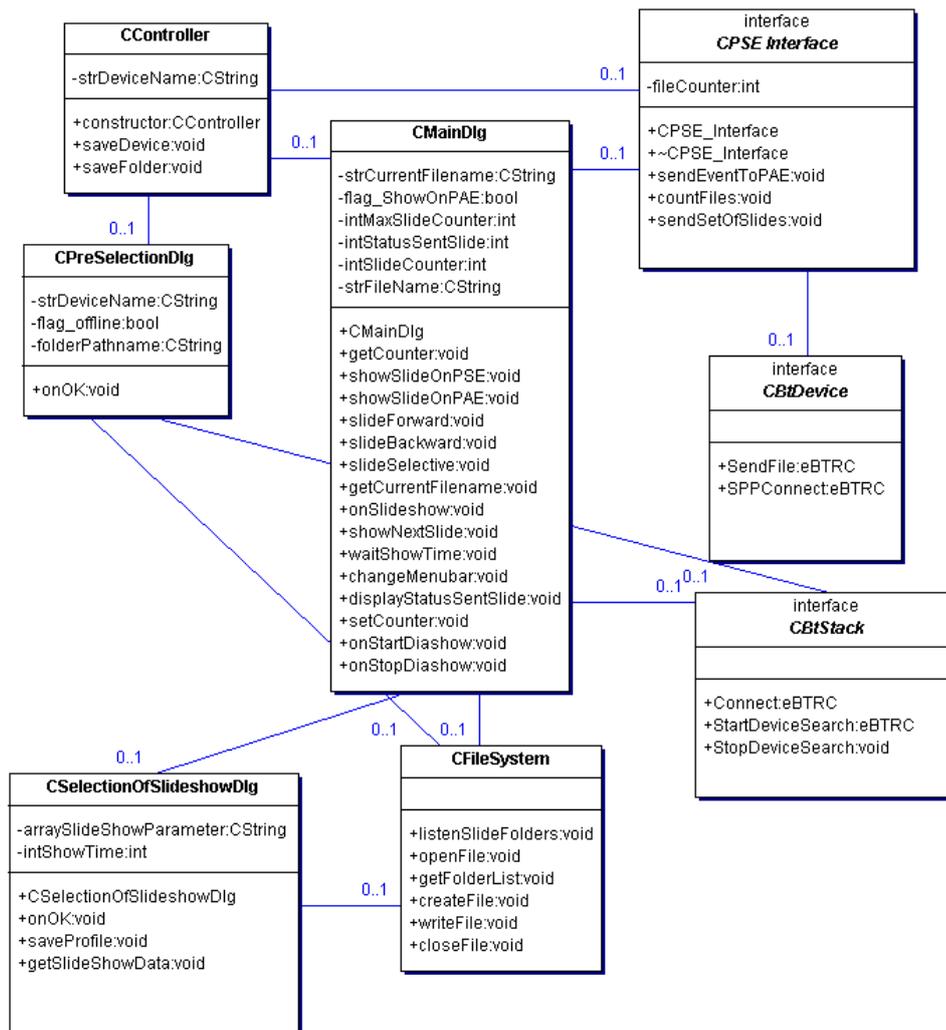


Abbildung 5.7: Klassendiagramm

Nachdem die einzelnen Sequenzdiagramme näher beschrieben wurden, ist in Abbildung 5.7 das daraus entstandene Klassendiagramm zu sehen. Einige Klassen wurden von bereits vorhandenen Klassen abgeleitet. Andere, für das Projekt spezifische Klassen, wurden selber entwickelt. Die Aufgaben diese Klassen und deren Methoden und wie sie im einzelnen zusammenarbeiten, ist im Kapitel 5.3.1 zu finden. Eine detaillierte Erklärung der einzelnen Methoden und Membervariablen finden sie im Anhang dieser Arbeit.

Systemkomponenten

Hier wird das Projekt nicht aus der Sicht des Anwenders wie im Kapitel 5.3.1, sondern aus der Sicht des Systems beschrieben. Dies wurde gemacht, um den Aufbau des Klassendiagramms und der Komponentenansicht zu verdeutlichen.

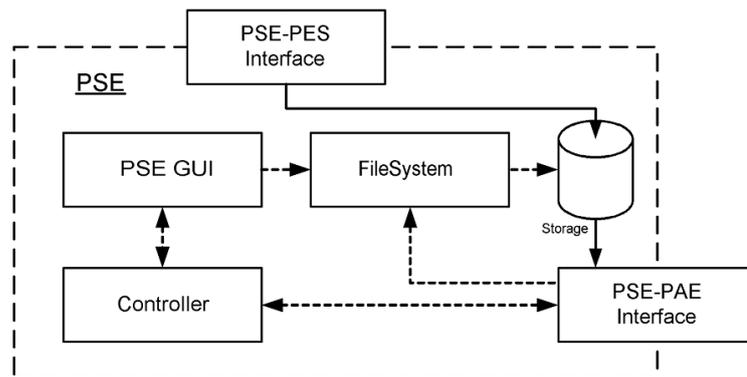


Abbildung 5.8: Systemkomponente PSE

Wie in Abbildung 5.8 ersichtlich, besteht die Systemkomponente PSE aus fünf einzelnen Komponenten, die alle unterschiedliche Aufgaben im Gesamtsystem wahrnehmen. Im weiteren Verlauf werden die einzelnen Komponenten mit ihren wichtigsten Methoden näher beschrieben.

- PSE GUI
- Controller
- FileSystem
- Interfaces
 - PSE-PES
 - PSE-PAE

PSE GUI

Die GUI-Komponente stellt die Verbindung zwischen dem Benutzer und dem System her. Sie ermöglicht es dem Benutzer in das System einzugreifen und es zu steuern. Die GUI

leitet lediglich die Eingaben des Benutzers zum Controller, bzw. bei einfachen Aufgaben auch gleich an die jeweiligen anderen Komponenten, weiter. Beispiele folgen in den jeweiligen Kapiteln der einzelnen Komponenten.

Die GUI besteht aus drei Klassen, die jeweils von der Oberklasse *CDialog* abgeleitet werden. Diese drei Instanzen lauten:

- CPreSelectionDlg
- CMainDlg
- CSelectionOfSlideshowDlg

Im weiteren Verlauf werden diese drei Dialog-Klassen, mit Hilfe ihrer wichtigsten Methoden, näher beschrieben, um einen Einblick in deren Aufgaben und Funktionsweise zu bekommen.



Abbildung 5.9: Vorauswahlfenster auf dem PDA

CPreSelectionDlg

Die Klasse *CPreSelectionDlg* erzeugt das in Abbildung 5.9 zu sehende Dialogfenster, welches sofort nach Programmstart zu sehen ist. Hier können die Voreinstellungen, wie Foliensatzauswahl, Bluetoothgerätauswahl oder Offlinebetrieb, vorgenommen werden. Diese Klasse an sich, stellt keine Methoden zur Verfügung. Sie dient lediglich dazu, die Voreinstellungen für das Programm vom Benutzer einstellen zu lassen und diese dann

zu übernehmen. Hierfür wird eine Foliensatzliste in eines der Auswahlfenster und eine Liste der gefundenen Bluetoothgeräte in dem anderen Fenster angezeigt. Es wird allerdings überprüft, ob der Benutzer einen Foliensatz ausgewählt hat und ob er ein Bluetoothgerät oder den Offlinebetrieb selektiert hat. Erst dann kann dieses Fenster vom Benutzer geschlossen und das Hauptprogramm gestartet werden.



Abbildung 5.10: Hauptfenster auf dem PDA

CMainDlg

Die CMainDlg-Klasse stellt dem Benutzer die in Abbildung 5.10 zu sehende Benutzeroberfläche zur Verfügung. In diesem Dialogfenster wird immer die aktuell ausgewählte Folie als Bild angezeigt. Ausserdem werden die Statusanzeigen für die bereits zur PAE übertragenden Folien (wenn Bluetooth-Gerät ausgewählt wurde) und für die momentan auf der PSE ausgewählte Folie angezeigt. Desweiteren ist es möglich über eine ComboBox eine Folie, die als nächstes angezeigt werden soll, auszuwählen (selektive Auswahl) oder über die Vor- bzw. Zurück-Tasten die vorherige bzw. nächste Folie auszuwählen (sequenzielle Auswahl). Zusätzlich kann aus der Menuleiste noch ein Untermenü geöffnet werden, aus dem man:

1. einen neuen Foliensatz öffnen kann
2. die Diashow starten kann
3. den Anzeige-Bestätigungs-Button de-/aktivieren kann
4. das Programm beenden kann

Die *CMainDlg*-Klasse stellt aber nicht nur die Benutzeroberfläche für das Programm zur Verfügung, sondern übernimmt noch folgende Aufgaben:

- Steuerung der Diashow (*changeMenubar*, *waitShowTime*, *showNextSlide* *onStartDiashow*, *onStopDiashow*)
- Steuerung der Präsentation (*slideForward*, *slideBackward*, *slideSelective*)
- Anzeige der Folien auf der PAE (*showSlideOnPAE*)



Abbildung 5.11: Diashowauswahl auf dem PDA

CSelectionOfSlideshowDlg

Diese Dialog-Klasse wird zur Laufzeit von *CMainDlg* erstellt. Sie bietet dem Benutzer, ein Dialogfenster an, indem die Voreinstellungen für die Diashow vorgenommen werden können (Siehe Abbildung 5.11).

Hier können folgende Einstellungen für die Diashow vorgenommen werden:

- Folienauswahl
- Anzeigedauer für die Folien
- Einstellungen als Profil abspeichern
- Profil laden

Controller

Die Controller-Komponente übernimmt im Programmablauf nur zwei Aufgaben: Bei der Folien- und Bluetoothgeräteausswahl sorgt sie dafür, dass der Name des Bluetoothgerätes (*strDeviceName* : *CString*) und der Ordnername des Foliensatzes (*folderPathName* : *CString*) in den jeweiligen Variablen abgespeichert wird. Dafür bietet sie die Methoden *saveDevice()* und *saveFolder()* an.

FileSystem

Diese Komponente besteht aus einer, von der „MFC CFile Class“, abgeleiteten Klasse und ist für die komplette Dateiverwaltung zuständig. Sie bietet zum einen Standard-Methoden, wie *createFile* *openFile* *writeFile* und *closeFile* zum Erstellen, Öffnen, Hineinschreiben und Schließen von Dateien an. Zum anderen stellt sie auch eigene Methoden, wie *listenSlideFolders()* zur Verfügung, mit der die Liste der vorhandenen Foliensätze in das Vorauswahlfenster geladen wird. Mit *getFolderList()* wird aus dem ausgewählten Foliensatzordner die einzelnen Folien im „Diashow Auswahlfenster“ angezeigt. Die Komponente FileSystem kommuniziert also in erster Linie mit den einzelnen Dialogklassen aber auch mit den Interfaceklassen, die im weiteren Verlauf noch erklärt werden.

Interfaces

Da die PSE zwei Schnittstellen hat, zum einen zum PES und zum anderen zur PAE, gibt es auch zwei Komponenten für diese Aufgaben. Diese Komponenten sind „PSE-PES Interface“ und „PSE-PAE Interface“.

PSE-PES

Diese Komponente hat die Aufgabe die Verbindung zum PES herzustellen. Da hierfür bereits ausreichende Programme zur Verfügung stehen, die unsere Anforderungen erfüllen, wurde auf eines dieser Tools zurückgegriffen. Die Folien werden mit Hilfe von „Active Sync“ in einen definierten Ordner auf der PSE übertragen und abgespeichert. D.h. das die eigentliche Arbeit von dem PES verrichtet wird, so dass hier keine weitere Software benötigt wird. Genauere Informationen über „Active Sync“ stehen im Kapitel 4.

PSE-PAE

Diese Komponente besteht aus drei unterschiedlichen Interface-Klassen.

- CBTStack
- CBTDevice
- CPSEInterface

BTAccess Klassen

Die Klasse „CPSEInterface“ wurde selber geschrieben, um die Verbindung zur PAE herzustellen und zu steuern. Die Klassen „CBTStack“ und „CBTDevice“ sind Klassen von der Firma High Point Software [19], die wir benutzen, um eine Verbindung zum Bluetooth-Stack des iPAQ 3870 herzustellen und um die Bluetooth-Profile in unserem Programm nutzen zu können. Weitere Informationen und eine genaue Auflistung der Methoden dieser Klassen entnehmen sie dem Kapitel 4.2.3.

CPSEInterface

Von dieser Klasse werden im Verlauf des Programms zwei Instanzen erzeugt.

- Streaming
- Event

Das „Streaming“-Objekt ist für das Versenden des Foliensatzes von der PSE zur PAE zuständig. Hierbei nimmt sie mit Hilfe der BTAccess-Klassen Verbindung zur PAE auf und schickt dann die Dateien alleine, Folie für Folie per Multimedia-Streaming, hinüber. Das „Event“-Objekt ist für das Versenden von Events zuständig. Die Events werden zuvor als „String“ in eine Datei geschrieben und dann zur PAE hinüberschickt. Wie dies im Detail funktioniert und welche Methoden dabei benutzt werden, ist im Kapitel 5.3.1 bereits beschrieben worden.

5.3.2 Hardware

Die Idee dieser Diplomarbeit basiert darauf, mit einem Gerät, welches man immer bei sich hat, ohne irgendwelche zusätzlichen Hilfsmittel, eine Foliensammlung (Präsentation) auf ein Anzeigegerät zu übertragen, anzeigen zu lassen und zu steuern. Das heisst dieses Gerät muss klein, energiesparsam, mit einer Funkverbindung (Bluetooth) und mit genügend Speicher ausgestattet sein, um dieses Programm darauf installieren zu können. Aufgrund dieser Fakten wurde ein PDA ausgewählt. Der Compaq iPAQ H3870 wurde wegen seines integrierten Bluetooth-Moduls bevorzugt.

Alternativ zum PDA könnte man sich auch ein Handy oder ein Web-Tablet vorstellen. Das Handy hat jedoch zur Zeit ein zu begrenztes Speichervolumen, um eine ausreichende Anzahl an Dateien (Folien) zu speichern und das Web-Tablet ist für eine Fernbedienung zu gross.

Detaillierte Informationen über den ausgewählten PDA und die benötigte Software sind im Kapitel 4.2 zu finden.

5.4 PAE - Präsentations-Anzeigeeinheit

Hier wird die dritte Systemkomponente beschrieben, die auch wieder in Software- und Hardware aufgeteilt wurde.

- Software
 - Sequenzdiagramm
 - Klassendiagramm
 - Systemkomponente
- Hardware
 - Notebook
 - Beamer

Der Software Bereich:

Als erstes wird ein Sequenzdiagramm gezeigt, welches die internen Abläufe beschreibt, die nötig sind, um eine vom Anwender gewünschte Folie auf dem Anzeigegerät sichtbar zu machen. Für das Streaming der Folien vom PSE zum PAE wurde kein Sequenzdiagramm entwickelt, da *TDK* die nötige Software hierfür schon bereitstellt und diese, ohne Anpassung, übernommen werden konnte.

Anschließend wird das Klassendiagramm beschrieben, welches aus zwei Klassen, einer GUI und einer Dateiverwaltung, besteht.

Nachfolgend soll die Systemkomponente PAE mit ihren Komponenten, PAE GUI, File-System und PAE Interface, beschrieben werden. Ein Bild zur Systemkomponente veranschaulicht die Beschreibung der Komponenten.

Der Hardware Bereich:

Hier soll erklärt werden, warum wir ein Notebook und ein Beamer benötigen.

5.4.1 Software

Sequenzdiagramm Folienanzeige PAE

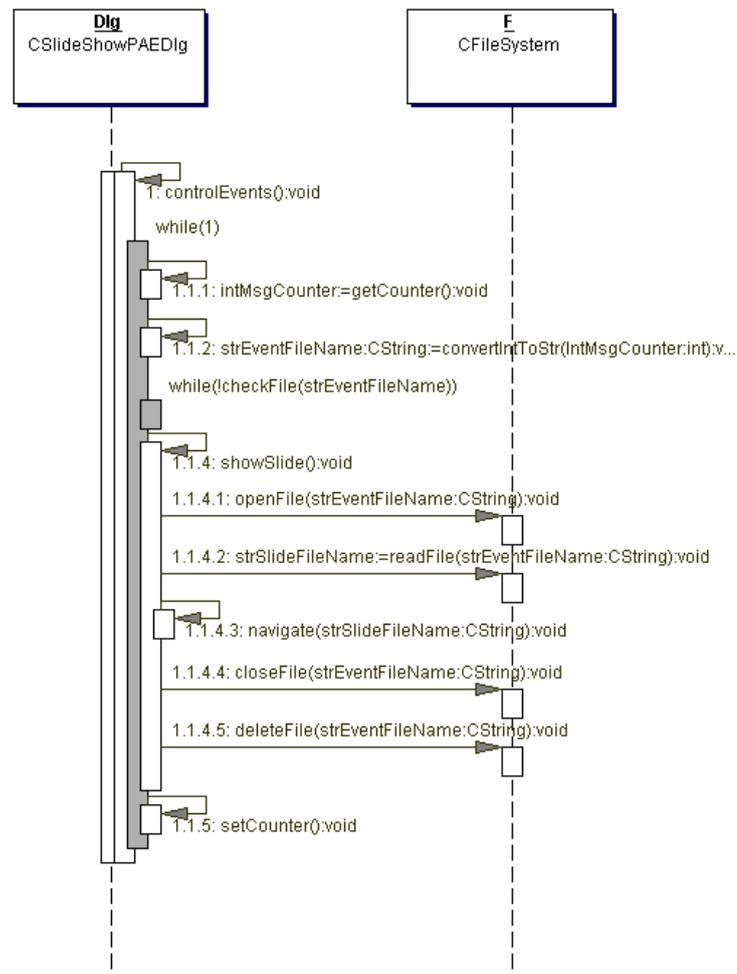


Abbildung 5.12: Anzeige der Folien auf dem Beamer.

Zur Veranschaulichung des hier folgenden Textes, bitte Abbildung 5.12 ansehen.

Zur Verdeutlichung der Interaktion zwischen der GUI- und FileSystem-Klasse und deren Findung, von weiteren vielleicht möglichen Klassen, sowie deren Methoden, wird hier ein Sequenzdiagramm gezeigt und erklärt.

In diesem Diagramm wird die TCP/IP Verbindung vom PSE zum PAE nicht mit erklärt, da die TDK-Software den Empfang der Folien und der Kontrolldateien per FTP übernimmt

und diese in einen definierten Ordner auf der PAE speichert. Es werden nur die Dateien, wenn benötigt, von der Dateiverwaltung aufgerufen. Weitere Informationen zum Streaming des Folien- und Kontroll-Dateien-Empfangs, siehe Erklärung Komponente: *PAE Interface* in Kapitel 5.4.1.

Beim Start des Programms auf dem Notebook wird die Methode `1:controlEvents()` der Klasse *CSlideShowPAEDlg* aufgerufen. In dieser Methode wird eine endlos-Schleife gestartet.

In einem FIFO-Ring, wird immer auf den nächsten Platz gewartet, bis er mit einer Kontroll-Datei belegt ist. Die Dateinamen der Kontrolldateien, werden mit 0 bis 9 benannt. Nach der Nummer 9, wird die nächste Datei mit 0 wieder von vorne benannt. Jede Aktion vom Anwender, eine neue Folie anzeigen zu lassen, wird mit einer Kontroll-Datei versehen. Diese wird von 0 bis 9 per Dateiname durchnummeriert und vom PSE zum PAE per FTP gesendet. Es wird kein großer Puffer benötigt, da nach Auslesen und Anzeigen der Folie auf dem Beamer die Datei wieder gelöscht wird und der Anwender in der Zeit höchst wahrscheinlich ein bis drei Aktionen, um weitere Folien anzeigen zulassen, ausführen wird.

Der nächste Platz des FIFO-Rings wird ausgelesen:

Mit Methode 1.1.1: `intMsgCounter:=getCounter()` wird die nächste Platznummer, 0 bis 9, des FIFO-Rings bestimmt.

Mit der Methode 1.1.2: `strEventFileName=convertIntToString()` wird die Platznummer zu einem Kontroll-Dateiname konvertiert.

Mit der `while(!checkFile(strEventFileName))` Schleife wird solange gewartet, bis der Platz in dem FIFO-Ring mit einer Kontroll-Datei gefüllt wurde.

Es wird die Methode 1.1.4: `showSlide()` aufgerufen, nachdem `checkFile == True` ist. Diese Methode sorgt dafür das:

1. Die Kontroll-Datei, mit 1.1.4.1: `openFile()`, geöffnet wird.
2. Diese wird mit 1.1.4.2: `readFile()` ausgelesen, um den Dateinamen der Folie zu bekommen, welche auf dem Beamer angezeigt werden soll.
3. Mit 1.1.4.3: `navigate()` wird die Folie im Browser angezeigt, somit auf dem Beamer sichtbar.
4. Mit 1.1.4.4: `closeFile()` wird die geöffnete Kontroll-Datei wieder geschlossen.
5. Zuletzt wird die Kontroll-Datei gelöscht und damit der Ring-Platz wieder freigegeben. Diese erledigt die Methode 1.1.4.5: `deleteFile()`

Nachdem die Methode 1.1.4: `showSlide()` beendet ist, wird mit 1.1.5 `setCounter()`, der Ring-Platz um eins inkrementiert und es kann von vorne losgehen.

Eine Fehlerbehandlung sowie das Beenden des Programms wurden in diesem Sequenzdiagramm nicht berücksichtigt.

Klassendiagramm PAE

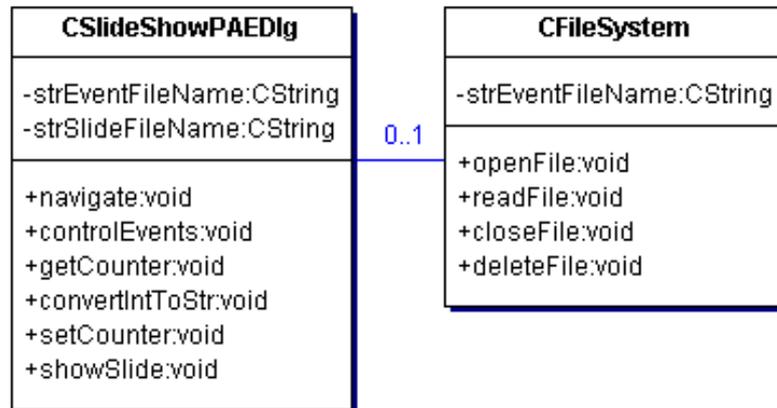


Abbildung 5.13: Klassendiagramm der PAE

Auch hier ist das in Abbildung 5.13 zu sehende Klassendiagramm aus dem zuvor beschriebenen Sequenzdiagramm entstanden. Die Aufgaben dieser Klassen und deren Methoden und wie sie im einzelnen zusammenarbeiten, ist im Kapitel 5.4.1 zu finden. Eine detaillierte Erklärung der einzelnen Methoden und Membervariablen finden sie im Anhang dieser Arbeit.

Systemkomponente

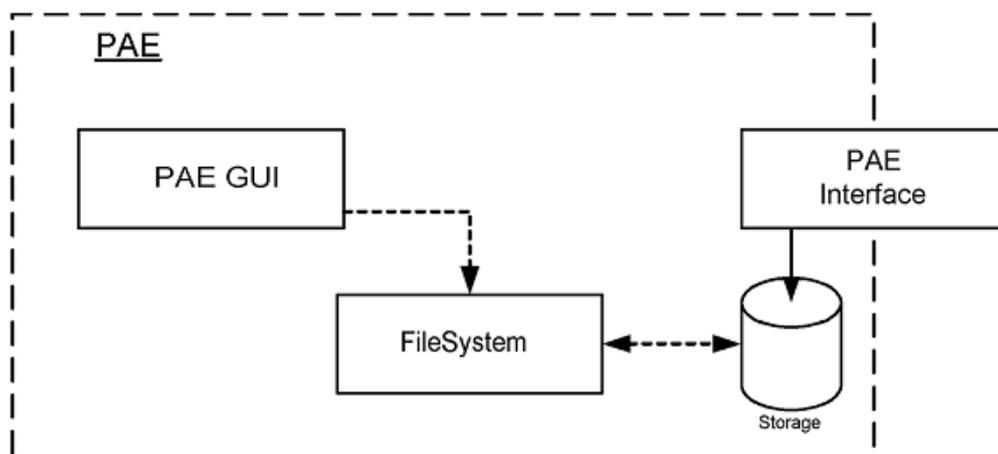


Abbildung 5.14: Systemkomponente der PAE

Die Systemkomponente PAE besteht, wie in Abbildung 5.14 zu sehen, aus drei einzelnen Komponenten. Die Aufgaben der einzelnen Komponenten und mit welchen Klassen und Methoden diese realisiert worden sind, wird in den folgenden Kapiteln näher beschrieben.

PAE GUI

Die Komponente *PAE GUI* wird mit der Klasse *CSlideShowPAEDlg* realisiert, die in diesem Fall keine GUI-Klasse im eigentlichen Sinne ist, da die „Benutzeroberfläche“ lediglich für die Anzeige der Folien zuständig ist. Dafür übernimmt sie die kompletten Steuerungsaufgaben für diese Komponente. D.h., sie steuert den Zugriff auf die „Event-Dateien“ mit Hilfe der *CFileSystem*-Klasse und verwaltet, die in Kapitel 5.4.1 bereits beschriebene, FIFO-Ring.

FileSystem

Diese Komponente besteht aus einer, von der „*MFC CFile Class*“, abgeleiteten Klasse und ist für die komplette Dateiverwaltung zuständig. Sie bietet der *CSlideShowPAEDlg*-Klasse die Standard-Methoden, wie *openFile*, *readFile*, *closeFile* und *deleteFile* zum Öffnen, Lesen, Schließen und Löschen von Dateien an. Die Komponente *FileSystem* stellt also nur Methoden für die Klasse *CSlideShowPAEDlg* zur Verfügung.

PAE Interface

Für die Interface-Komponente wurde nichts eigenes entwickelt, da es bereits Anwendungen auf dem Markt gibt, die die Anforderungen erfüllen. Diese Komponente muss in der Lage sein, Dateien, die über eine Bluetooth-Verbindung zur PAE geschickt werden, zu empfangen und in einen definierten Ordner zu speichern. Mit der Installation des Treibers für den Bluetooth-USB-Adapter von TDK wird automatisch auch der BTTRAY mit installiert. Dieses Programm ermöglicht es dem Benutzer, in der Nähe befindliche Bluetooth Geräte zu finden und eine Verbindung mit ihnen aufzunehmen. Mit diesem Programm ist es also möglich, der PAE Dateien über eine TCP/IP Verbindung zu schicken und diese in einem bestimmten Ordner abzuspeichern. Den Speicherort kann man in den Einstellungen für dieses Programm ändern.

5.4.2 Hardware

Für diese Komponente wird eine Hardware benötigt, die es möglich macht, Dateien über Bluetooth zu empfangen, diese auf einem Speichermedium zu speichern und auf einer für alle Zuhörer sichtbaren Anzeigeeinheit zu präsentieren.

Beamer

Da es nötig ist die Folien so groß wie möglich anzuzeigen und da in vielen Räumen, in denen Präsentationen abgehalten werden (z.B. Vorlesungsräume in der HAW-Hamburg) bereits Vorkehrungen für die Verwendung solcher Geräte getroffen wurden, wurde für diesen Zweck ein Beamer ausgewählt. Für diese Arbeit wird der in der HAW-Hamburg zur Verfügung stehende Beamer benutzt. Eine genaue Beschreibung dieses Gerätes ist im Kapitel 4.3.2 zu finden. Heute gibt es zwar auch schon die ersten Beamer mit integriertem Bluetooth, die jedoch den finanziellen Rahmen dieser Diplomarbeit sprengen würden. Über eine Verwendung dieser Geräte für diese Arbeit wird aber im Anhang berichtet.

Notebook

Dadurch, dass der Beamer kein Prozessor, sowie genügend Speicher besitzt und desweiteren keine integrierte Bluetooth-Schnittstelle hat, muss also noch ein Gerät an den Beamer angeschlossen werden, welches einen VGA-Ausgang hat und über eine Bluetooth-Empfangseinheit verfügt. Hierfür wird ein Notebook verwendet, welches mit einem Bluetooth-USB-Adapter von TDK ausgestattet ist. Die Daten für das Notebook und den Bluetooth-USB-Adapter sind im Kapitel 4.3 nachzulesen. Eine weitere Möglichkeit wäre ein „VGA-Dongel“. Dieser müsste über eine Bluetooth-Empfangseinheit mit ausreichend Speicher und über einen VGA-Ausgang verfügen. Weitere Informationen sind im Anhang zu finden.

5.5 Zusammenfassung

In diesem Kapitel wurde über das Software-Design und über die Realisierung der Software gesprochen.

Jede einzelne Komponente wurde dabei einzeln betrachtet und ausgewertet. Für jede Komponente, für die zuvor die Anforderungen ermittelt und eine Analyse erstellt wurden, ist hier das Design entworfen und ein erster Realisierungsansatz entwickelt worden. Dabei wurde das System aus zwei unterschiedlichen Perspektiven betrachtet.

Zum einen aus der Sicht des Anwenders. Dies wurde mit Sequenzdiagrammen beschrieben, bzw. durch die Dialogfenster, indem die Interaktion zwischen dem System und dem Anwender dargestellt und beschrieben wurden.

Zum anderen aus der Sicht des Systems. Durch die Darstellung und Beschreibung der einzelnen Komponenten der drei Systemkomponenten und des Klassendiagramms wurde das System ohne Einwirkung des Anwenders betrachtet und beschrieben.

Die drei Systemkomponenten (PES, PSE und PAE) wurden jeweils in Hard- und Software aufgeteilt.

Hardware:

Im Hardware-Design wurde nach Geräten gesucht, die die gestellten Anforderungen erfüllen. Dazu wurde nach geeigneten Geräten gesucht, diese wurden verglichen und bewertet. Anschließend wurde das am besten geeignete Gerät ausgewählt und genauer beschrieben.

Software:

Bei einigen Komponenten ist im Design festgestellt worden, dass es bereits geeignete Programme gibt, die den Anforderungen genügen und somit nicht mehr realisiert zu werden brauchen. Für alle anderen Anforderungen wurde ein Software-Design durchgeführt und erste Realisierungsansätze erläutert.

Kapitel 6

Resümee

Dieses Kapitel beschäftigt sich mit dem entstandenen Design und dessen Realisierung. Die Diplomarbeit wird mit anderen, ähnlichen Projekten verglichen. Desweiteren werden Stärken und Schwächen aufgezeigt, sowie Verbesserungs- und Erweiterungsvorschläge gemacht.

6.1 Erreichte Ziele

Unser Hauptziel, eine Fernbedienung, zur Steuerung der Folienanzeige und zum Transport der Folien, zu entwickeln, die von seiner Handhabung leicht bedienbar ist, haben wir erreicht. D.h., mit unserem „Easy Presenter“ ist es möglich nach Betreten des Vortragsraums sofort mit der Präsentation zu beginnen.

Desweiteren haben wir folgende Ziele erreicht:

Eine Fernbedienung (bis zu 10 m Entfernung) zu entwickeln, welche kabellos (über Bluetooth Funkübertragung) und ohne Sichtkontakt zur Anzeigeeinheit funktioniert.

Wir haben außerdem das Ziel, bezüglich der Änderbarkeit des Designs in Hinsicht auf folgende Bereiche, erreicht:

1. Die Wahl der Software für die Erstellung der Präsentation.
2. Die Wahl des Dateiformates für die Folien, des zu übertragenden Datei-Formates, wie jpg, bmp, xml, html, usw.

3. Die Wahl der Übertragungstechnologie wie Bluetooth, WLAN, IRDA usw. Mit der kleinen Einschränkung, dass bei der Wahl von WLAN der PDA ausgetauscht werden müsste, da dieser nicht WLAN fähig ist, siehe hierzu Kapitel: „Ausblick“.
4. Die Austauschbarkeit der Hardwarekomponente für die Fernbedienung wie z.B.: PDA, Weblet, Handy, usw.
5. Die Austauschbarkeit der Hardwarekomponente für die Anzeigeeinheit wie z.B.: nur Beamer mit VGA-Dongel ohne Notebook oder Beamer, wie der Sharp PG-M25x mit WLAN-Fähigkeit, ohne extra VGA-Dongel. Siehe Erklärung im Kapitel: „Ausblick“.

Die Durchführung dieser Arbeit stellte uns vor die Schwierigkeiten, mit einer komplett neuartigen Funk-Übertragungs-Technologie: (*hier Bluetooth*) und der Entwicklungsumgebung für embedded Visual C++ die ersten Schritte zu gehen, welches an unserer Hochschule noch nicht vorhanden war. Diese Arbeit soll als eine Grundlage für nachfolgende Projekte im Bereich Bluetooth in dieser Richtung gelten. Die Anfänge waren sehr schwierig, da es zum Beginn unsere Diplomarbeit noch keine fehlerfreie Software sowie Treiber für die Bluetooth Übertragung von Daten gab. Erst mit den Updates dieser Software und Treiber war es möglich, eine Verbindung über Bluetooth aufzubauen und Daten von einem Rechner zum anderen zu übertragen. Desweiteren ist ein Betriebssystem Windows 2000 oder höher notwendig gewesen, um eine Netzwerkverbindung über Bluetooth zu erreichen. Die Informationen hierfür, waren zu unserem Zeitpunkt der Diplomarbeit noch sehr rar zu bekommen.

Ein Problem für unsere Diplomarbeit stellt der Verbindungsaufbau über Bluetooth dar. Wenn ein PDA als Fernbedienung mit dem Notebook als Anzeigegerät das erste mal in Verbindung treten möchte, muss aus Sicherheitsgründen eine feste Bindung zwischen den Geräten eingegangen werden. Dieses muss zur Zeit noch manuell vom Anwender erledigt werden. Das bedeutet, das der Anwender mit dem PDA erst nach anderen Bluetooth Geräten suchen, dann das Gerät auswählen und dann die feste Bindung mit dem anderen Gerät eingehen muss. Hierbei wird ein „Request“ mit einem „PIN“, der vom Anwender eingegeben wurde, zum Notebook gesendet. Am Notebook muss dieser „Request“, mit dem richtigen PIN bestätigt werden. Dieser Vorgang muss nur einmal durchgeführt werden, da es sich hier um eine feste Verbindung (Partnerschaft) handelt. Einen Lösungsvorschlag für dieses Problem zeigen wir im Kapitel „Ausblick -> weitere Ergänzungen“ auf.

6.2 Ausblick

In diesem Kapitel wird beschrieben, was aus heutiger Sicht verbessert oder sogar ergänzend zur unserer Diplomarbeit dazu kommen könnte.

6.2.1 Andere Fernbedingung

Aus heutiger Sicht würden wir den COMPAQ iPAQ H3870 gegen den HP iPAQ H5550 austauschen, da der neue PDA zusätzlich auch noch die WLAN Funkübertragung zur Verfügung stellt. Dadurch würde die Wahl der Übertragungstechnologie auf Hinsicht der Änderbarkeit keine Rolle mehr spielen. Desweiteren bietet der neue PDA noch einen Fingerprint, dadurch wäre die Sicherheit noch mehr gewährleistet.

6.2.2 VGA-Dongel

In unserer Diplomarbeit besteht die Anzeigeeinheit aus 2 Geräten, das erste ist das Notebook und das zweite der Beamer. Eine sinnvolle Erweiterung für eine weitere Diplomarbeit wäre die Einsparung des Notebooks. Es müßte eine Art VGA-Dongel für den Beamer entwickelt werden. Dieser VGA-Dongel braucht folgende Komponenten:

1. **Bluetooth Empfänger-/Sendeeinheit.**
Z.B. ein kleiner USB-Bluetooth-Adapter am VGA-Dongel per USB-Schnittstelle anschließbar.
2. **Speicher**
für die Folien-Dateien, Bluetooth-Software, Server-Software für Anzeige der Folien, usw.
3. Ein Prozessor wird benötigt.

6.2.3 Beamer mit WLAN

Eine weitere Möglichkeit das Notebook als Anzeigeeinheit einzusparen, bietet eventuell der neue Beamer von Sharp PG-M25x [18]. Dieser Beamer ist WLAN-fähig und arbeitet mit den meisten Programmen wie Powerpoint, Excel und Word zusammen.

6.2.4 Multimedistreaming

Eine Anforderung haben wir in unserer Diplomarbeit noch nicht gestellt. „Was passiert, wenn der Anwender z.B. von Seite 1 gleich auf Seite 25 springen möchte und sie ist noch nicht geladen?“ Die Erfüllung dieser Anforderung, die gewünschte Seite sofort zu übertragen, wäre eine sinnvolle Erweiterung unserer Diplomarbeit. Bis jetzt ist es so, dass die Steuereinheit vom PSE nachsieht, ob die Folie Nr. 25 schon übertragen wurde. Wenn dieses nicht passiert ist, wird dem Anwender eine Meldung, wie „Bitte warten..“ auf

dem PDA-Display, ausgegeben. Im schlimmsten Fall müssen erst alle Folien übertragen werden, bis die Folie angezeigt werden kann und das könnte unnötig Zeit in Anspruch nehmen.

Daher wäre eine sinnvolle Erweiterung, nicht nur die Meldung anzeigen zu lassen, sondern das Multimedia-Streaming kurz zu unterbrechen, um eine Folie gesondert (per FTP) übertragen zu können und diese in einer gesonderten Liste als übertragen zu markieren. Wenn jetzt wieder eine Folie aufgerufen wird, fragt die Steuereinheit einmal die per Streaming übertragenden Folien ab. Wenn das fehlschlägt, wird zusätzlich die gesonderte Liste abgefragt. Weitere Anregungen zu dem Thema: „Multimedia Streaming - Vergleich Video Streaming im Internet“ [8].

6.2.5 Weitere Ergänzungen

- Präsentation empfangen und speichern
Die auf dem Home PC erstellte Präsentation soll auf dem PDA empfangen und abgespeichert werden können. Dies soll ein eigenständiges Programm auf dem PC erledigen. Zurzeit müssen die Folien-Dateien noch manuell über die *ActiveSync* Software in den richtigen Ordner auf dem PDA übertragen werden.
- Informationsfenster zur selektierten Folie anzeigen.
In einem Textfenster werden zusätzliche Notizen zur selektierten Folie angezeigt. Diese Notizen sind editierbar.
- Die Reihenfolge der Folien in einem Foliensatz soll änderbar sein und in dieser Reihenfolge soll per Multimediasstreaming die einzelnen Folien-Dateien übertragen werden können.
- Feste Bindung zwischen zwei Geräten, mit PIN Austausch, in die Software auf dem PDA integrieren.
Hierfür werden in den BTACCESS Klassen die nötigen Methoden angeboten. Dies wird aber nicht vollautomatisch laufen, da der Anwender den „Request“ vom PDA immer noch mit der Eingabe des „PINs“, auf dem Notebook, bestätigen muss. Da auf der Seite des Notebooks keine BTACCESS Klassen einsetzbar sind und uns keine weiteren Möglichkeiten zum heutigen Zeitpunkt vorliegen. Eine andere Möglichkeit zur automatischen Bindung zwischen zwei Bluetooth Geräten, ohne Vernachlässigung der Sicherheit, bietet Sony mit der neuen FEEL-Technik. Diese Technik setzt auf Bluetooth 1.1 auf und kommt, laut Sony, ohne zusätzliche Hardware aus. Die automatische Bindung wird durch sehr kurze Distanzen, von zwei bis drei Zentimeter, zwischen beiden Bluetooth Geräten, ausgelöst. Durch diese kurzen Distanzen, werden die Sicherheitsaspekte gewährleistet. Die automatische Bindung dauert ca. 5 Sekunden. Weitere Informationen siehe: „Das Fünf-Sekunden-Programm“ [7].

Anhang A

Beschreibung der Methoden und Variablen

A.0.6 CPreSelectionDlg

Memberfunktionen

OnOK

Wird aufgerufen, wenn der Benutzer nach Beendigung seiner Voreinstellungen den OK-Button gedrückt hat. Danach werden die vom Benutzer ausgewählten Voreinstellungen vom Programm übernommen.

Membervariablen

strDeviceName:CString

Name des ausgewählten Bluetoothgerätes

flag_offline:bool

Wird auf TRUE gesetzt, wenn der Offline-Betrieb ausgewählt wurde, d.h. wenn die Präsentation ohne Bluetoothgerät durchgeführt werden soll

folderPathname:CString

Ordnername des ausgewählten Foliensatzes

A.0.7 CMainDlg

Memberfunktionen

showSlideOnPSE

Die geladene Folie wird auf der PSE angezeigt.

showSlideOnPAE

Die auf der PSE angezeigte Folie wird auf PDA angezeigt. Dazu muss diese zuvor zur PAE übertragen worden sein.

slideForward

Aktuelle Folie wird um eins inkrementiert und auf der PSE angezeigt.

slideBackward

Aktuelle Folie wird um eins dekrementiert und auf der PSE angezeigt.

slideSelective

Anwender wählt selektiv eine bestimmte Folie aus, um diese anzeigen zu lassen.

getCurrentFilename

Erstellt den Dateinamen der aktuell anzuzeigenden Folie. Der Dateiname wird anschließend in der Variablen strFileName abgespeichert.

onSlideshow

Wird ausgeführt, wenn der Benutzer den Slideshow-Modus ausgewählt hat. Danach gelangt man in das Diashow-Vorauswahl-Fenster. Es wird also die Klasse CSelectionOfSlideshowDlg erstellt.

showNextSlide

Ist eine Memberfunktion, die in der CMainDlg selbst aufgerufen wird, um die nächste Folie auf der PSE zu laden und anzeigen zu lassen.

waitShowTime

In dieser Funktion wird solange gewartet, wie vorher in intShowTime (Anzeigedauer der Folien), gespeichert wurde.

changeMenubar

Ändert am Ende der Diashow die Menubar am unteren Rand des Bildschirms wieder um, in die des CMainDlg.

displayStatusSentSlide

Diese Methode sorgt dafür, dass die Anzahl der bereits zur PAE übertragenden Folien auf dem Display der PSE angezeigt wird. Die Nummer der zuletzt übertragenden Folie holt sich diese Methode dabei aus der Variable fileCounter.

setCounter

Diese Methode erhöht jedesmal, wenn eine Folie übertragen wurde, die fileCounter Variabel.

getCounter

Mit dieser Methode wird die Platznummer des FIFO-Rings ermittelt.

onStartDiashow

Wird ausgeführt, wenn der Benutzer das Vorauswahlfenster geschlossen hat und mit der Präsentation beginnen will.

onStopDiashow

Wird ausgeführt, wenn der Benutzer den „Stop“-Button während der Diashow drückt. Die Methode sorgt dafür, dass die intSlideCounter Variable auf den Wert der letzten Foliennummer gesetzt wird, so dass die Präsentation danach wieder von vorne beginnt.

Membervariablen**strCurrentFilename:CString**

Dateiname der aktuelle Folie die gerade auf dem PSE zu sehen ist und auf dem Beamer angezeigt werden kann.

flag_ShowOnPAE:bool

Dieses Flag wird gesetzt, wenn der Anzeigebutton aktiviert worden ist. True == AnzeigebestätigungsbUTTON muß gedrückt werden, um die Folie auf dem PAE anzeigen zu lassen. False == Folie wird automatisch auf dem PDA und Beamer angezeigt!

intMaxSlideCounter:int

Gibt die vorher ermittelte Gesamtanzahl der Folien des Foliensatzes an. Dies wird gebraucht, um z.B. bei slideForward das Ende des Foliensatzes zu erkennen.

intStatusSentSlide:int

Gibt die Anzahl der schon übertragenen Folien vom PDA zum Notebook an. Dieser Wert wird auf der PSE angezeigt, wenn die Folien zur PAE übertragen werden.

intSlideCounter:int

Nummer der aktuellen Folie, die gerade auf der PSE angezeigt wird.

strFileName:CString

Aktueller Dateiname, der gerade beim Streaming übertragen wird.

A.0.8 CSelectionOfSlideShowDlg

Memberfunktionen

onOK

Wird ausgeführt, wenn der Benutzer den OK-Button im Diashow-Vorauswahl-Fenster gedrückt hat. Danach wird das Vorauswahlfenster geschlossen und das Hauptfenster wieder angezeigt.

saveProfile

Wird ausgeführt, wenn der Benutzer das Diashow-Profil abspeichern will, also den „Save“-Button gedrückt hat. Als nächstes wird überprüft, ob ein Profil-Name eingegeben wurde (filenameLength!=0) und ob Folien für die Diashow ausgewählt wurden (ShowFileListLength!=0).

getSlideShowData

Der Benutzer hat in der rechten Listbox, seine Angaben über die Folienauswahl sowie deren Reihenfolge gemacht. Diese Einstellungen werden in einem Array gespeichert(arraySlideShowParameter). Ausserdem wird die vom Benutzer eingestellte Anzeigedauer übernommen. Diese Anzeigedauer wird in der Variable intShowTime gespeichert.

Membervariablen

arraySlideShowParameter:CString

Der Benutzer hat in der rechten Listbox seine Angaben über die Folienauswahl sowie deren Reihenfolge gemacht. Diese Einstellungen werden in dem Array `arraySlideShowParameter` gespeichert.

intShowTime:int

Dies ist der Parameter, der bei der Diashow-Konfiguration angegeben werden muss, um die Dauer der Anzeige je Folie festzulegen.

A.0.9 CController

Memberfunktionen

saveDevice

Mit dieser Methode wird das zuvor ausgewählte Bluetoothgerät in der Variablen (`strDeviceName:CString`) abgespeichert. Wenn das `flag_offline` auf TRUE ist, wird als Gerätenamen NULL eingesetzt.

saveFolder

Mit dieser Methode wird der Ordnername des zuvor ausgewählten Foliensatzes in der Variablen (`folderPathName:CString`) abgespeichert.

Membervariablen

strDeviceName:CString

Name des zuvor im Vorauswahlfenster(Folien- und Bluetoothauswahl) ausgewählten Bluetoothgerätes.

A.0.10 CFileSystem

Memberfunktionen

listenSlideFolders

Die Liste der vorhandenen Foliensätze (Ordnernamen) wird in das Vorauswahlfenster geladen.

getFolderList

Mit *getFolderList()* wird aus dem ausgewählten Foliensatzordner die einzelnen Folien im „Diashow Auswahlfenster“ angezeigt.

createFile

Erstellt eine neue Datei.

openFile

Öffnet eine vorhandene Datei.

writeFile

Schreibt Daten in eine vorhandene Datei.

closeFile

Schließt eine Datei.

A.0.11 CPSEInterface**Memberfunktionen****sendEventToPAE**

Baut eine Verbindung zum PAE auf, sendet die „Event-Datei“, in der der „Event-String“ gespeichert wurde, zur PAE, um die Folien anzeigen zu lassen und baut die Verbindung wieder ab.

countFiles

Zählt die zur PAE übertragenden Dateien. Wird gebraucht, um festzustellen, ob alle Folien zur PAE übertragen worden sind.

sendSetOfSlides

Ist zuständig für die Übertragung der einzelnen Folien zum PAE und des Setzen des „fileCounters“ für die schon übertragenen Folien.

Memervariablen

fileCounter:int

Wird immer, wenn eine Datei mit der Methode *sendSetOfSlides()* übertragen wurde, um eins inkrementiert. Zeigt also die Anzahl der bereits zur PAE übertragenden Folien an.

A.0.12 CBTStack

General:	
<i>Connect:</i>	Initialize a connection to the stack
<i>Disconnect:</i>	Terminate a connection to the stack
<i>StartDeviceSearch:</i>	Start searching for Bluetooth devices
<i>StopDeviceSearch:</i>	Stop searching for Bluetooth devices
<i>GetActiveConnections:</i>	Get list of devices to which you are currently connected
<i>GetVersion:</i>	Get version string of the BTAccess library
Security:	
<i>GetGeneralSecurityOptions:</i>	Get/Set values for Device name, Default Passkey, Device-discoverable, Device-connectable, Transmit Power level, and FTP shared directory
<i>SetGeneralSecurityOptions:</i>	Get/Set values for a specific service, including Service-enabled, Authorization required, PINCode required, and Encryption Required
<i>GetSvcSecurityOptions:</i>	Get/Set values for a specific service, including Service-enabled, Authorization required, PINCode required, and Encryption Required
<i>SetSvcSecurityOptions:</i>	
Device Group (*BTAccess only):	
<i>GetDeviceGroups:</i>	Get list of all Device Group names
<i>GetActiveDeviceGroup:</i>	Get the currently active Device Group
<i>SetActiveDeviceGroup:</i>	Select the Device Group to be active
Local Device (*BTAccess only):	
<i>GetBtAddress:</i>	Get Bluetooth address of the local device
<i>IsBtRadioOn:</i>	Test whether Bluetooth radio is on or off
<i>BtRadioOn:</i>	Turn the Bluetooth radio on
<i>BtRadioOff:</i>	Turn the Bluetooth radio off
<i>IsBtBuiltIn:</i>	Test whether the device has built-in Bluetooth hardware
<i>IsBtExpansionPackPresent:</i>	Test whether the Bluetooth Expansion Pack is inserted
<i>SoftReset:</i>	Perform a soft reset of the iPAQ

Tabelle A.1: Methoden der Klasse: *CBtStack*

A.0.13 CBTDevice

<i>SendBusinessCard:</i>	Send my business card to this device
<i>GetBusinessCard:</i>	Get this device's business card
<i>ExchangeBusinessCards:</i>	Exchange business cards with this device
<i>SendFile :</i>	Send a single file to this device
<i>CancelSendFile:</i>	TCancel a pending SendFile operation
<i>GetServiceList:</i>	Get list of service names for particular service type (typically only for serial service, which can have multiple named instances)
<i>SPPConnect:</i>	Connect over BT serial port to this device
<i>SPPDisconnect:</i>	Disconnect serial port
<i>DUNConnect:</i>	Connect using BT Dial-up Networking port to this device
<i>DUNDisconnect:</i>	Disconnect Dial-up Networking
<i>LAPConnect:</i>	Connect to a Lan Access Point
<i>LAPDisconnect:</i>	Disconnect from Lan Access Point
<i>PANConnect:</i>	Connect to this device for PAN (Personal Area Network) access, as client or host node
<i>PANDisconnect:</i>	Disconnect from this device for PAN access
<i>CreateBond:</i>	Create pair-bond with this device
<i>RemoveBond:</i>	Remove pair-bond with this device
<i>IsBonded:</i>	Test whether this is device is bonded or not
<i>SetAutoConnect:</i>	Configure this device for autoconnect using serial port, Dial-up Networking, or LAN Access
<i>GetSignalStrength:</i>	Get RSSI signal strength of any connection to this device from the local device (low, good, high)

Tabelle A.2: Methoden der Klasse: *CBtDevice*

A.0.14 PAE GUI

Memberfunktionen

navigate

Zeigt die Folie im Browser an und macht sie somit auf dem Beamer sichtbar.

controlEvents Wird bei Programmbeginn gestartet und läuft in einer Endlosschleife. Von hier aus werden alle anderen Methoden aufgerufen.

getCounter

Holt die nächste Platznummer, aus der *intMsgCounter*-Variablen

convertIntToStr

Konvertiert die Platznummer zu einem Kontroll-Dateinamen

setCounter

Inkrementiert den Ring-Platz um eins.

showSlide

Von hier aus werden alle weiteren Methoden aufgerufen, die für die Dateiverwaltung und für das Anzeigen der Folien zuständig sind.

checkFile

Überprüft, ob eine Kontroll-Datei in der FIFO-Ring vorhanden ist.

Membervariablen**intMsgCounter**

Gibt den Platz der aktuellen Kontroll-Datei in dem FIFO-Ring an.

strEventFileName:CString

Ist der Name der aktuellen Kontroll-Datei.

A.0.15 CFileSystem**Memberfunktionen****openFile**

Öffnet eine Datei

readFile

Liest eine Datei aus

closeFile

Schließt eine Datei

deleteFile

Löscht eine Datei

Anhang B

FAQ

Da wir auch einige Anfangsschwierigkeiten mit dem Umgang des iPAQ H3870 sowie des Bluetooth-USB-Adapter am Notebook hatten, wollen wir an dieser Stellen unsere Lösungen zu diesen Schwierigkeiten anbieten:

- Beim Starten von ActiveSyn auf dem PDA über USB-Dockstation, stürzte der PDA immer ab!

Lösung:

Hardware Reset behebt kurzzeitig das Problem. Um es langfristig zu lösen, benötigt man ein Betriebssystem-Update: IPAQ Pocket PC 2002 Update 1 Version 1.00 08.01.2002

- Verbindung von PC->PDA über ActiveSyn per bluetooth funktioniert nicht!

Lösung:

1. Neuen Bluetooth Service Manager für IPAQ H3xxx installieren.
 2. Neuen Treiber für den Bluetooth-USB-Adapter(TDK) installieren.
 3. Betriebssystem Pocket 2002 Update
- Das Fenster „BT Network Access“ wird auf dem PDA nicht minimiert.

Lösung:

Unter Windows 98 haben wir das Problem nicht gelöst bekommen!

Unter Windows XP und 2000 läßt sich das Problem mit einigen Einstellungen beheben.(siehe CD)

- Das Fenster „BT Network Access“ wird auf dem PDA jetzt zwar minimiert. Aber es ist uns noch nicht möglich gewesen, einen Ping zum PDA oder vom PDA zum PC zu schicken.

Lösung:

Unter Windows 2000 ist es jetzt auch möglich, einen Ping von PDA zum PC und in die andere Richtung zu senden (nähere Beschreibung mit Screenshots siehe CD)

Anhang C

Installationshinweise

Um eine Bluetooth-Verbindung zwischen dem iPAQ H3870 und dem Bluetooth-USB-Adapter am Notebook herzustellen, mußte folgende Software auf unserem Notebook installiert werden:

1. ActiveSync 3.1
2. Bluetooth-USB-Adapter Treiber für TDK (Version 1.2)
3. Jetzt erst den Bluetooth-USB-Adapter am PC anschließen!!!
4. Entwicklungsumgebung für PDA: Microsoft eMbedded Visual Tools 3.0
5. PocketPC 2002 SDK (Emulator und Bibliotheken)

Um eine problemlose Übertragung per Bluetooth, vom iPAQ H3870 zum Notebook mit Bluetooth-USB-Adapter, zu gewährleisten, sind folgende Treiber Updates notwendig:

1. Betriebssystem
IPAQ Pocket PC 2002 Update 1
Version 1.00 08.01.2002
2. IPAQ H3xxx Bluetooth Service Manager
Version 1.1 18.04.2002
3. Bluetooth-USB-Adapter
Treiber Update für den PC
Version 1.2 26.02.2002
4. iPAQ H3800 Series Driver Updates
Version 1.00 (25 Mar 02)

Anhang D

Glossar

ACL(Asynchronous Connectionless)

Folgende Erklärung des Begriffs: ACL, wurde von INTEREST¹ - Lexikon IT-Fachbegriffe, übernommen:

Abk. Asynchronous Connectionless. Oberbegriff für asynchrone verbindungslose (connectionless) Datenübertragungen. Beispiel: ACL-Übertragung im Nahbereichsfunksystem Bluetooth. [21]

Folgende Erklärung des Begriffs: ACL, wurde von „Bluetooth“, von Nico Edtinger, übernommen:

Eine Verbindung bei Bluetooth kann entweder synchron und verbindungsorientiert (SCO) oder asynchron und verbindungslos (ACL) sein.

SCO (synchronous connection-oriented link) ist eine Verbindung zwischen einem Master und einem Slave, wobei der Master regelmäßige Zeitlots reserviert um eine bestimmte Bandbreite garantieren zu können. Dies kann für Echtzeitanwendungen wie Sprache verwendet werden, wobei allerdings nicht auf Datenintegrität geprüft wird. Dies fällt allerdings nicht so sehr ins Gewicht da die Hauptanforderung eben die Echtzeitfähigkeit ist.

ACL-Pakete werden in Timeslots gesendet die nicht für eine SCO-Verbindung vorgesehen wurden. ACL (asynchronous connectionless link) ist eine Verbindung zwischen Master und einem oder mehreren Slaves. Zwischen einem Slave und dem Master kann immer nur eine ACL-Verbindung bestehen. Die

¹ist ein Geschäftsbereich der WEKA MEDIA GmbH & Co. KG mit Sitz in Kissing.

Variante mit mehreren Slaves ist für Broadcasts gedacht. Ausserdem kann eine ACL-Verbindung die Datenintegrität prüfen und bei Fehler die Daten erneut übertragen. [20]

baseband protocol (Basisband-Protokoll)

Folgende Erklärung des Begriffs: baseband protocol, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Baseband gehört zu den Bluetooth-Kernprotokollen und liegt im Bluetooth-Schichtenmodell unterhalb der Link Manager-Schicht mit dem Link Manager Protocol (LMP). Das Protokoll dieser Schicht realisiert die HF-Verbindungen in einem Piconet. Die paketvermittelte HF-Übertragung arbeitet nach dem Frequenzsprung-Verfahren (FHSS) mit definierten Zeitschlitzten. Bluetooth unterscheidet zwischen der asynchronen verbindungslosen (ACL) Übertragung und der synchronen verbindungsorientierten (SCO). [22]

Bluetooth

Folgende Erklärung des Begriffs: Bluetooth, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Bluetooth ist der Standard für die Funk-Kommunikation mit geringen Reichweiten, die bei etwa 10 Metern liegen und durch die festgelegte Sendeleistung von 0 dBm bedingt ist. Durch Einsatz von Verstärkern kann die Entfernung auf 100 Meter erhöht werden. Bluetooth wurde ursprünglich entwickelt, um mit Hilfe der Funktechnik die vielen Kabelverbindungen zwischen Geräten abzuschaffen. Im Gegensatz zu der IrDA-Technik können die Bluetooth-Geräte dabei auch ohne Sichtkontakt miteinander kommunizieren und neben Daten kann auch Sprache übertragen werden. Der Bluetooth-Standard hat das Ziel, die Kurzstrecken-Kommunikation zwischen bis zu acht Endgeräten wie Notebooks, Organizer, PDAs und Handys zu unterstützen. Aber auch die Fernsteuerung von Druckern, Fernsehern, Radios oder anderen elektronischen Geräten ist vorgesehen.

Mittels Bluetooth können kleine Wireless LANs aufgebaut werden, in denen viele LAN-relevante Anwendungen durchgeführt werden können. Andererseits ist auch eine Sprachkommunikation zwischen Bluetooth-Geräten aber auch über eine funktechnische Anbindung an das ISDN oder das analoge Fernsprechnetzt möglich. Ein weiteres Einsatzgebiet könnte die Steuerung von Haushaltsgeräten vom Notebook aus sein. [22]

<http://www.bluetooth.com>

Bluetooth-Netz

Folgende Erklärung des Begriffs: Bluetooth-Netz, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Mittels Bluetooth, dem Standard für mobile Kommunikation im Nahbereich, können bis zu acht Geräte in einem so genannten Piconet miteinander kommunizieren. Bis zu 10 Piconets lassen sich zu einem Kommunikationsverbund vereinen. Mehrere Piconets mit sich überlappenden Funkbereichen werden als Scatternet bezeichnet. Dabei kann der Slave in einem anderen Piconet sein. Es gibt verschiedene Betriebsarten. So den Parkmode bei asynchronen verbindungslosen Diensten (ACL) oder den Sniffmode.

Bluetooth-Netzkonfiguration: Ein Scatternet bestehend aus zwei Piconets Gesteuert wird das Bluetooth-Netz von einem Master, wobei innerhalb eines Piconets eine Bluetooth-Einheit die Rolle des Masters übernimmt, während die anderen Einheiten als Slaves arbeiten. An aktiven Einheiten kann eine Pico-Funkzelle einen Master und maximal sieben Slaves haben. Zusätzlich können bis zu 255 passive Slaves, im so genannten Parkmodus eingebucht sein. Diese passiven Slaves synchronisieren sich laufend mit dem Master. Dieser kann sie aus dem Parkmodus in den Kommunikationsstatus versetzen. Der Master bestimmt mittels eines Zufallsgenerators die Abfolge der Frequenzsprünge, die die Teilnehmer eines Piconets verwenden. [22]

CGI (common gateway interface)

Folgende Erklärung des Begriffs: CGI, wurde von INTEREST² - Lexikon IT-Fachbegriffe, übernommen:

Durch die Programmierschnittstelle CGI ist es möglich, Programme auf einem Server abzulegen, die von einem Browser gestartet werden können. Solche Programme (oder Scripts) können z. B. Formulareingaben aus HTML-Dateien verarbeiten, auf dem Server Daten speichern und dort gespeicherte Daten auslesen. Die CGI-Schnittstelle besteht aus:

- einem Verzeichnis auf dem Server-Rechner, das CGI-Programme enthalten darf, und
- einer Reihe von Umgebungsvariablen, die teilweise vom Server unabhängig vom CGI-Programm mit Daten belegt werden (z. B. mit Angaben zum Server selbst oder zum Browser, der zuletzt eine Anfrage gestartet hat).

²ist ein Geschäftsbereich der WEKA MEDIA GmbH & Co. KG mit Sitz in Kissing.

Auf die CGI-Umgebungsvariablen kann mit verschiedenen Programmiersprachen zugegriffen werden. Wenn man eine Compiler-Sprache benutzt, muss man allerdings dafür sorgen, dass das Programm für die Betriebssystemumgebung des Servers kompiliert und gelinkt wird. Neben der CGI-Schnittstelle gibt es andere, von kommerziellen Herstellern erstellte Schnittstellen. Von Netscape stammt z. B. die NSAPI-Schnittstelle, von Microsoft die ISAPI-Schnittstelle. [21]

Folgende Erklärung des Begriffs: CGI, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Das Common Gateway Interface (CGI) ist ein standardisiertes, plattformunabhängiges Interface, das den Informationsaustausch zwischen Web- oder HTTP-Servern und einem externen Programm regelt. CGI ist somit der Standard zur Ausführung externer Programme von WWW-Servern. Über das Common Gateway Interface können andere Systeme, wie Datenbanken, mit HTTP-Servern kommunizieren. CGI-Programme müssen für den Datenaustausch den CGI-Spezifikationen entsprechen, wobei die Programme dank der Plattformunabhängigkeit in beliebigen Programmiersprachen wie C, Perl oder Java, geschrieben werden können. Benutzereingaben, wie beispielsweise der Klick auf einen Hyperlink, werden über den Browser abgeschickt und vom Web-Server mittels CGI an das auszuführende Programm weitergeleitet. Nach der Bearbeitung gibt das Programm die Ergebnisdaten mittels CGI an den Server zurück. [22]

CORBA (common object request broker architecture)

CORBA ergab sich aus der Standardisierung des ORB.

Folgende Erklärung des Begriffs: CORBA, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Das im Jahre 1989 durch ein Firmenkonsortium (OMG) entwickelte CORBA bildet eine Referenzarchitektur für verteilte, objektorientierte Systeme für die Objektvermittlung, wobei die Portabilität und Interoperabilität von Software-Komponenten einen wesentlichen Entwicklungsaspekt bildet. CORBA beschreibt die Interaktion zwischen Client und Server in einer Client-Server-Architektur, bei der Objekte über einen offenen Software-Bus zusammenarbeiten. Basis dieser Architektur bildete die Veröffentlichung der Object Management Architecture (OMA). Eine entscheidende Komponente von

COBRA ist der Object Request Broker (ORB), der die Vermittlungsfunktion zwischen den Objekten übernimmt. Der ORB wurde als Common Object Request Broker Architecture (CORBA) standardisiert. Mit der CORBA 2.0-Spezifikation gibt es einen Standard, der die Kommunikation zwischen heterogenen Plattformen verschiedener Hersteller unterstützt. Mit diesem Standard können Objekte auch dann miteinander kommunizieren, wenn sie von verschiedenen Herstellern stammen oder auf unterschiedlichen Betriebssystemen arbeiten. [22]

HTML (hypertext markup language) Folgende Erklärung des Begriffs: HTML, werden von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

HTML ist die Beschreibungssprache für Dokumente im World Wide Web. Sie stellt eine Teilmenge der Programmiersprache SGML dar und enthält die Sprachelemente für den Entwurf von Hypertext-Dokumenten. Um die Komponenten, die Hierarchie und die Verknüpfungspunkte der Dokumente zu beschreiben, benutzt HTML diverse Codes. HTML kann die Hypertext-Verbindungen über die ganze Welt ausdehnen. Sie kann die Informationen an die jeweilige Benutzeroberfläche anpassen und bildet eine Experimentierplattform für Hypermedia. HTML kann für die Darstellung von Menüs benutzt werden, aus denen Optionen gewählt werden können, ebenso zur Darstellung von Ergebnissen von Datenbankabfragen, zur Dokumentation und für Online-Hilfstexte. [22]

Jini (java intelligent network infrastructure)

Sun nennt es „spontaneous networking“. In Jini werden sowohl Daten als auch Code über das Netz bewegt, alles unter der strengen Typisierung von Java. Konsequenterweise wird bei Jini alles (also Hard- und Software) als „Objekt“ angesehen und letztendlich gleich behandelt. Jini dient dabei als Abstraktionsstufe und vereinigt die Sichten auf unterschiedliche Komponenten.

L2CAP (logical link control and adaptation protocol)

Folgende Erklärung des Begriffs: LLCAP, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Das Logical Link Control and Adaptation Protocol (L2CAP) ist ein Bluetooth-Kernprotokoll das das Multiplexing und die Segmentation and Reassembly-Funktionen (SAR) übernimmt. L2CAP verbindet die höheren Schichten des

Protokollstacks mit dem Link Manager-Protokoll (LMP) und parallel dazu mit dem Baseband-Protokoll. L2CAP kann über RFCOMM und die darüber liegenden Transportprotokolle verbindungslose und verbindungsorientierte Dienste anbieten. In L2CAP können Dienstgüten (QoS) spezifiziert werden, und zwar betreffend den Datenfluss nach RFC 1363, dessen Verzögerung und Verzerrungen. [22]

LMP (link manager-protocol)

Folgende Erklärung des Begriffs: LMP, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Das Link Manager Protocol (LMP) gehört wie L2CAP, SDP und Baseband zu den Bluetooth-Kernprotokollen. Dieses Protokoll bildet die Schnittstelle zu dem Host Controller Interface (HCI) und verbindet die höheren Schichten des Protokollstacks mit den übertragungstechnischen Baseband-Protokoll. Neben dem Verbindungsaufbau handelt das Link Manager Protocol die Paketgröße für das Baseband-Protokoll aus, steuert die HF-Sendeleistung und übernimmt Sicherheitsfunktionen in Form einer Authentifizierung und Verschlüsselung mittels Sicherheitscodes. Bei der Authentifizierung wird ein 128-Bit-Schlüssel auf der Verbindungsstrecke eingesetzt. [22]

MAC-Adresse

Hardware-Adresse eines jeden LAN-Controllers, die zur eindeutigen Identifikation eines Knotens im Netzwerk dient. Die MAC-Adresse wird fest in einem Chip eingebrannt und kann nicht mehr verändert werden.

OMG (object management group)

Object Management Group ist ein Zusammenschluß von Hardwareherstellern, Softwareentwicklern, Netzbetreibern und kommerziellen Nutzern von Software. Kern der Arbeit der OMG ist eine Architektur für die Verteilung und Kooperation objektorientierter Softwarebausteine in heterogenen, vernetzten Systemen.

Folgende Erklärung des Begriffs: OMG, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Die Object Management Group (OMG) hat in den 90er Jahren plattformübergreifende Standards für die Anwenderintegration erarbeitet, die bei den

Funktionsaufrufen der programmiersprachlichen Objekte ansetzen. Ein Beispiel für die Anwenderintegration ist CORBA, das in einem Middleware-Standard direkt in Web-Services einsetzbar ist. [22]

<http://www.omg.org>

ORB (object request broker)

ORB bildet die Basiskomponente für die Kommunikation zwischen verschiedenen Services. Er dient dazu, Client/Server-Beziehungen zwischen Objekten aufzubauen.

Folgende Erklärung des Begriffs: ORB, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Verteilte Objekte sind dezentralisierte, auf dem Adressraum von mehreren Systemen abgelegte Objekte, die mit den Objekten auf den unterschiedlichen Systemen zusammenarbeiten. Die für die Zusammenarbeit notwendige Vermittlerrolle übernimmt der Object Request Broker (ORB). ORBs lassen Objekte, die auf viele Rechner verteilt sind, effizient zusammenarbeiten und gaukeln ihnen vor, dass sie sich im selben Adressraum befinden. Der ORB verfügt über eine Liste aller zugänglicher Komponenten und reicht Aufgaben an den richtigen Bearbeiter weiter. ORB wurde von der Object Management Group (OMG) unter CORBA standardisiert und übernimmt die Vermittlungsfunktion, in dem er Anfragen an Objekte entgegen nimmt, die Objekte lokalisiert und die Anfrage und das Ergebnis übermittelt. Mögliche ORB-Implementierungen sind in verschiedenen Implementierungsklassen festgelegt: Client- und Implementierungsresidente ORB, Server-basierte ORB, System-basierte ORB und Bibliotheks-basierte ORB. [22]

PAE (Präsentations-Anzeige-Einheit)

Die Präsentationsanzeigeeinheit ist für die Anzeige der Folien auf dem Beamer verantwortlich. Die Steuerung erfolgt von der PSE aus, lediglich die gewünschte Folie wird auf dem Notebook, bzw. Beamer angezeigt.

PES (Präsentations-Erstellungssystem)

Das Präsentations-Erstellungssystem, ist wie der Name schon aussagt, für die Erstellung einer Präsentation, sowie Konvertierung der einzelnen Folien in das Format: .jpg, zuständig.

PSE (Präsentations-Steuer-Einheit)

Die Präsentationssteuereinheit ist für die Steuerung des Präsentationsablaufs zuständig. Der Anwender kann mit dem PSE interagieren und somit den Ablauf der Präsentation steuern.

RFCOMM (RF communication)

RFCOMM ist einer Emulation, seriellen Schnittstelle, gleichzusetzen.

Eine weitere Erklärung des Begriffs: RFCOMM, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Das RF Communication-Protokoll (RFCOMM) ist ein Bluetooth-Protokoll, das über die Luftschnittstelle die serielle Verbindung einer RS-232-Schnittstelle mit ihren Steuer- und Datensignalen emuliert. Dieses Protokoll wird auch als »Cable Replacemant Protocol« bezeichnet. RFCOMM kann L2CAP-Verbindungen multiplexen und maximal 60 RS-232-Verbindungen bedienen. Die RFCOMM-Schicht, auf der Netzwerkprotokolle wie IP, Routing-Protokolle wie PPP und Transportprotokolle wie TCP oder UDP aufsetzen, ermöglicht die drahtlose Kommunikation von derzeitigen und zukünftigen Anwendungen. Dabei laufen die Anwendungen über eine oder mehrere vertikale Kanäle des Protokollstacks. RFCOMM basiert auf Standards der ETSI TS 07.10 und ITU Q.931. [22]

RMI (remote method invocation)

Fern Methoden Aufruf in Java Eine weitere Erklärung des Begriffs: RMI, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

RMI ist eine objektorientierte Java-Auflage des Remote Procedure Calls, RPC. Das Remote Methode Invocation konkurriert mit dem plattformübergreifenden Cobra-Standard. Mittels RMI kann ein Client transparent serverseitige, in einem Java-Interface definierte Methoden aufrufen. [22]

RPC (remote procedure call)

Aufruf von Prozeduren auf entfernten Netzteilnehmern

Eine weitere Erklärung des Begriffs: RPC, wurde von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

RPC ist ein von SUN für NFS entwickeltes Protokoll für die Ebenen 5 und 6 und gewährleistet einen entfernten Funktionsaufruf. Jeder Server im Netz stellt im Rahmen dieses Konzeptes eine Anzahl von Diensten zur Verfügung, die mit RPC angefordert werden können. Diese Funktionen sind als Prozeduren eines Programms realisiert und können unter Angabe von Serveradresse, Programmnummer und Prozedurnummer angesprochen werden. RPC ist in den RFCs 1057 und 1831 beschrieben. [22]

SAR (segmentation and reassembly sublayer)

Folgende Erklärung des Begriffs: SAR, von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Segmentierungs- und Wiedervereinigungsteilschicht

Teilschicht der ATM-Anpassungsschicht. Die SAR-Teilschicht ist verantwortlich für die Segmentierung der Informationseinheiten der höheren Schichten in ATM-Zellen sowie die Zusammensetzung von Zelleninhalten. Die SAR erhält von der Konvergenz-Teilschicht die PDUs und fügt den einzelnen Datenblöcken einen Header und einen Trailer hinzu. Die neu entstandenen PDUs heißen SAR-PDUs. Diese werden am Service Access Point (SAP) zur ATM-Schicht übergeben, wo sie als ATM-SDU den Payload für die ATM-Zellen bilden. Bei dem Prozess in der SAR-Teilschicht werden Frames mit variabler Länge in Zellen mit fester Länge umgewandelt sowie die umgekehrte Funktionalität ausgeführt. [22]

SCO(Basisband-Protokoll)

Folgende Erklärung des Begriffs: SCO, wurde von „Bluetooth“, von Nico Edtinger, übernommen:

Eine Verbindung bei Bluetooth kann entweder synchron und verbindungsorientiert (SCO) oder asynchron und verbindungslos (ACL) sein.

SCO (synchronous connection-oriented link) ist eine Verbindung zwischen einem Master und einem Slave, wobei der Master regelmäßige Zeitlots reserviert um eine bestimmte Bandbreite garantieren zu können. Dies kann für Echtzeitanwendungen wie Sprache verwendet werden, wobei allerdings nicht auf Datenintegrität geprüft wird. Dies fällt allerdings nicht so sehr ins Gewicht da die Hauptanforderung eben die Echtzeitfähigkeit ist.

ACL-Pakete werden in Timeslots gesendet die nicht für eine SCO-Verbindung vorgesehen wurden. ACL (asynchronous connectionless link) ist eine Verbindung zwischen Master und einem oder mehreren Slaves. Zwischen einem

Slave und dem Master kann immer nur eine ACL-Verbindung bestehen. Die Variante mit mehreren Slaves ist für Broadcasts gedacht. Ausserdem kann eine ACL-Verbindung die Datenintegrität prüfen und bei Fehler die Daten erneut übertragen. [20]

SDP (service discovery protocol)

Folgende Erklärung des Begriffs: SDP, werden von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Das Service Discovery Protocol (SDP) gehört zu den Bluetooth-Kernprotokollen. Das Protokoll dient der Erkennung der Bluetooth-fähigen Geräte untereinander, damit diese miteinander kommunizieren können. Über dieses Protokoll werden die Verbindungen zwischen den Bluetooth-Geräten eingerichtet und alle für die Datenübertragung relevante Informationen übermittelt. Beim SDP-Protokoll startet der SDP-Client eine Anfrage (SDP-Request) an den SDP-Server, der ihm einen SDP-Response schickt. [22]

Stubs (common object request broker architecture)

Stubs, lokale Stellvertreter für die entfernten Objekte und der Umwandlung der Parameter für den Transport (Remote Procedure Call)

TCS (telephony control specification)

Folgende Erklärung des Begriffs: TCS, werden von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Bei den Telephony Control Specification (TCS) handelt es sich um die Telefon-APIs. In den TCS wird die Signalisierung zur Rufabwicklung spezifiziert, damit Sprach- und Datenverbindungen zwischen Bluetooth-fähigen Geräten aufgebaut werden können. Das TCS kann durch TCS-Binary oder durch AT-Befehle realisiert werden. TCS-Binary ist ein Bit-orientiertes Protokoll mit Rufkontrolle, Verbindungsaufbau, Sprach- und Datenübertragung. Die AT-Befehle dienen der Steuerung für Handys und Modems und machen auch eine Fax-Übertragung möglich. Die AT-Befehle sind vom standardisierten AT-Befehlssatz abgeleitet. [22]

UDP Folgende Erklärung des Begriffs: UDP, werden von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

user datagram protocol (UDP)

Das User-Datagram-Protokoll ist ein Transportprotokoll (Schicht 4) des OSI-Referenzmodells und unterstützt den verbindungslosen Datenaustausch zwischen Rechnern. Das UDP wurde definiert um auch Anwendungsprozessen die direkte Möglichkeit zu geben, Datagramme zu versenden und damit die Anforderungen transaktionsorientierten Verkehrs zu erfüllen. UDP baut direkt auf dem darunter liegenden IP-Protokoll auf. [22]

WLAN (wireless lokal area network)

Folgende Erklärung des Begriffs: WLAN, werden von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

Wireless LANs sind Lokale Netze, die über Funk oder Infrarotlicht arbeiten, also ohne Kabel. Sie sind standardisiert unter dem IEEE-Standard 802.11. Die Übertragung wird entweder mit Funkfrequenzen im Mikrowellenbereich oder mit Infrarotlicht durchgeführt. Bei Multicast- und Broadcast-Anwendungen existieren in Abhängigkeit von der Übertragungsgeschwindigkeit zwei Leistungsklassen: Low & Medium Speed mit bis zu 1 Mbit/s und High Speed mit über 1 Mbit/s. Entsprechend der genutzten Übertragungsfrequenz unterscheidet man die drahtlosen Netze in Spread-Spectrum-Netze (DSSS), die Funkübertragung mit dem Frequenzsprung-Verfahren (FHSS) und die Schmalband-Mikrowellen-Netze. Spread-Spectrum ist die am weitesten verbreitete Technologie und liegt im Frequenzbereich von 2,4 GHz. In diesem Frequenzband stehen zwischen 60 bis 80 Kanäle mit einer Übertragungsrate von 1 Mbit/s zur Verfügung. Für die schnelleren Übertragungsraten von 11 Mbit/s stehen nur drei Kanäle zur Verfügung. Die Schmalbandtechnik nutzt vorwiegend den Frequenzbereich von 5,775 GHz, aber auch den Schmalbandmikrowellen-Bereich von 18,8 GHz bis 19,2 GHz. Die Anwendung beider Technologien unterliegt damit gesetzlichen Bestimmungen.

Übersicht über die Übertragungsbereiche von WLANs

Die Radiosysteme mit Spread-Spectrum oder Mikrowelle können zwei unterschiedliche Topologien haben: die Sterntopologie oder die Bustopologie. Bei der Sterntopologie agiert ein zentraler Hub als Schaltstelle im LAN, beim Bussystem sendet jede Station unmittelbar an alle anderen. Bei allen

drahtlosen Systemen wird die Gesamtleistung maßgeblich von ihrer physikalischen Umgebung bestimmt. Generell sind bei dem Einsatz von drahtlosen LANs bautechnische und physikalische Gegebenheiten, die die Übertragung und die Ausdehnung der LANs beeinträchtigen, zu berücksichtigen. Spread-Spectrum-Systeme können Distanzen bis zu 300 m überwinden. Bei den drahtlosen Netzen besteht darüber hinaus noch ein Zusammenhang zwischen dem erreichbaren Durchsatz und der maximalen Entfernung zwischen den Knoten: je größer die Entfernung, desto kleiner der Durchsatz. Jedes drahtlose Netzwerk wird in Funkzellen eingeteilt. Eine Funkzelle besteht im Minimum aus einem Sender/Empfänger-Paar und wird als der Raum definiert, in dem alle Sender und Empfänger dieselbe Frequenz und/oder denselben Code benutzen. Je nach Ausdehnung der einzelnen Funkzellen unterscheidet man zwischen Pico-Funkzellen, Mikro-Funkzellen und Makro-Funkzellen.

Datenübertragungsraten für unterschiedliche Zellengrößen

WLANs kennen zwei Betriebsarten, den Infrastructure-Mode und den Ad-Hoc-Mode. Beim Infrastructure-mode stellt das Funknetz den Zugang zum einem kabelbasierten lokalen Netz her. Im Ad-Hoc-Betrieb wird spontan eine direkte Verbindung zu den Netzteilnehmern innerhalb des WLANs aufgebaut. [22]

XML (extended markup language)

Folgende Erklärung des Begriffs: XML, werden von Siemens Aktiengesellschaft Information and Communication Networks, übernommen:

XML ist die Standardsprache zur Definition von individuellen Markup-Sprachen, die strukturierte Informationen auszeichnen sollen. Bei XML handelt es sich um eine Untermenge des ISO-Standards SGML; diese wurde 1998 vom W3C als Metasprache akzeptiert. XML kann auf verschiedenen Betriebssystemen und unterschiedlichen Anwendungen eingesetzt werden. Sie stellt ebenso wie HTML eine verknüpfungsorientierte Sprache der Web-Inhalte dar, sie stellt eine allgemeine Syntax bereit, um hierarchisch strukturierte Daten zu beschreiben. Inhalte und ihre Präsentation lassen sich getrennt darstellen. XML fungiert als Auszeichnungssprache mit der neue Sprachen beschrieben werden können. XML-Dokumente können sowohl von Menschen als auch vom Computer gelesen werden. Die Sprache erleichtert den Austausch von Daten zwischen verschiedenen Unternehmen bei E-Commerce, B2B- und B2C-Anwendungen. Bei XML obliegt die Seitengestaltung dem Surfer. [22]

Anhang E

Evolution

E.1 TCP/IP Verbindung

E.1.1 Versuch: Senden eines Textstrings

Objektorientierter Ansatz:

Abgeleitet von der Klasse: `cAsyncSocket`

Verwendete Methoden: `create`, `connect`, `listen`, `accept`

Von PC zum PC funktionierte diese Netzwerkverbindung einwandfrei, aber von PDA zum PC gab es folgendes Problem: Die Methoden `create`, `connect` und `listen` wurden fehlerfrei ausgeführt, nur bei der Methode `accept` ist der PDA hängen geblieben. Wir haben das Problem nicht weiter analysiert und haben uns aus Zeitgründen für einen funktionsorientierten Ansatz entschieden.

Weitere Informationen über den Objektorientierten Ansatz finden sie auf der Homepage:

<http://www.mut.com/media/buecher/VCPLUS6/data/kap20.htm>

Funktionsorientierter Ansatz:

Funktionen aus der Headerdatei: `Winsock.h`

Verwendete Funktionen: `bind()`, `socket()`, `listen()`, `accept()`, `connect()`, `send()`, `recv()`

Verwendete Funktionen auf der Serverseite:

```
WSAStartup( MAKEWORD ( 2, 2 ), &wsaData ); //für Fehlermeldung
serverSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
```

```

struct sockaddr_in server;

memset(&server, 0, sizeof(server));
server.sin_family = AF_INET;
server.sin_port = htons( 4000 );
server.sin_addr.s_addr = INADDR_ANY;
rc = bind(serverSocket, (sockaddr*)&server, sizeof(sockaddr_in));
listen( serverSocket, SOMAXCONN )
newSocket = accept(serverSocket, (sockaddr*)&client, &n);
fRet = CheckForRecv(newSocket);
bytes = send( newSocket, test, 99, 0 );
bytes = recv( newSocket, recvd, 99, 0 );
rc = closeSocket(serverSocket);

```

Verwendete Funktionen auf der Clientseite:

```

WSAStartup( MAKEWORD ( 2, 2 ), &wsaData ); //für Fehlermeldung
Socket = socket(AF_INET, SOCK_STREAM, 0);
server.sin_family = AF_INET;
server.sin_port = htons( 4000 );
server.sin_addr.s_addr = inet_addr("92.168.18.131");
rc = connect(socket, (SOCKADDR*) &server, sizeof(SOCKADDR_IN))
rc = send(socket, fileBuffer, sizeof(fileBuffer), 0);
rc = recv(socket, rvBuffer, sizeof(rvBuffer), 0);
rc = closesocket(socket);

```

Dieser Ansatz funktioniert fehlerfrei. Das Senden und Empfangen von Strings ist hiermit möglich. Desweiteren mussten wir das Senden und Empfangen von Dateien überprüfen.

E.1.2 Versuch: Senden einer Folien Datei

Funktionsorientierter-Ansatz:

Beim ersten Versuch haben wir einen funktionsorientierten Ansatz verwendet. Hierfür haben wir folgende Funktionen aus der Headerdatei stdio.h benutzt:

```

//gefordertes File von Platte lesen
if ((stream = fopen(path,"rb"))!= NULL){
//einlesen der Datei in fileBuffer
//char fileBuffer[50000];
flen = fread( fileBuffer, sizeof(char), sizeof(fileBuffer), stream);
fclose(stream);
}

```

Problem:

Wenn der „fileBuffer > 50000 (ca. 50kB)“ deklariert wird, stürzt das Programm auf der PDA Seite ab. Wir vermuten, daß ein Datenspeicher „> 64kB“ auf PDA-Seite zum Absturz führt. Da wir aber auch Dateien über „50kB“ übertragen müssen, mußten wir hierfür eine andere Lösung finden:

Objektorientierter-Ansatz:

Beim zweiten Versuch haben wir einen objektorientierten Ansatz für die Dateiübertragung verwendet.

Wir haben uns überlegt, die Datei nicht im Ganzen zu verschicken, sondern in Blöcken einzulesen und zu versenden. Dadurch ist die fileBuffer-Größe auf 4kB geschrumpft. Hierfür haben wir folgende Methoden aus der Klasse CFile benutzt.

PDA-Seite:

```
CFile sourceFile;
sourceFile.Open(sourcePath, CFile::modeRead | CFile::shareDenyWrite, &ex)
do {
// Einlesen von 4096-byte Blöcken
// merken wie viel Bytes schon eingelesen wurden
dwRead = sourceFile.Read(fileBuffer, 4096);
// aktuell eingelesener Block senden
bytes = send( newSocket, fileBuffer, 4096, 0 );
// die Schleife endet, wenn alle Blöcke ausgelesen sind
}while (dwRead == 4096);
    sourceFile.Close();
```

PC-Seite:

```
CFile destFile;
destFile.Open(destPath, CFile::modeWrite |
CFile::shareExclusive | CFile::modeCreate, &ex)
do {
bytes = recv( newSocket, fileBuffer, 4096, 0 );
destFile.Write(fileBuffer, bytes);
}while (bytes == 4096);
destFile.Close();
```

E.2 ActiveX Browser in Visual C++ einbinden

Wir haben uns für ActiveX Control entschieden, da der „Microsoft Web Browser“ als ActiveX Control entscheidene Vorteile, gegenüber anderen Varianten, für uns aufweist. Ein wichtiger Vorteil für die Änderbarkeit der Diplomarbeit, ist hiermit durch die Anzeige von verschiedenen Datei-Formaten möglich, die vom ActiveX Control akzeptiert werden, wie z.B.: html, htm, xml, jpg, bmp, png, usw.

Ein weiterer Vorteil ist die einfache Handhabung.

Hier ein Beispiel, wie dieses ActiveX Control im Projekt mit eingebunden wird:

1. Neues Projekt erstellen:
 - Location: C:/TEMP/test
 - MFCAppWizard(exe)
 - Step 1 Dialog based
 - Step 2 - 4 ActiveX Controls
2. Resources Karte in der Registry auswählen, um das Dialog-Fenster öffnen zu können.
3. Mit der rechten Maustaste auf die Dialogbox -> insert ActiveX Control -> „Microsoft Web Browser“ auswählen.
4. Wieder rechte Maustaste -> ClassWizard -> Member Variables -> *IDC_EXPLORARI* -> ADD Variable (Klasse wird erzeugt. Einfach mit „OK“ bestätigen und danach Name für MemberVariabel = *m_Browser* eingeben.)
5. Message Maps -> *IDCANCEL* -> *BN_CLICKED* -> EDIT CODE
6. Diese Zeile in die Funktion kopieren: *m_browser.Navigate(„http://www.web.de“,0,0,0,0);*
7. „RebuildALL“ und Start der „execute“
8. „CANCEL“ Button drücken, jetzt wird erst die Website angezeigt!!!

Hier ein Beispiel, wie die Größe des anzuzeigenden Bildschirms eingestellt wird und eine Datei, z.B. aus dem Internet mit „html“ Format geladen wird.

```
// TODO: Add extra initialization here
SetWindowPos(&wndTopMost, 0, 0, 1200, 800, NULL);
m_browser.SetTop(0);
m_browser.SetLeft(0);
m_browser.SetWidth(1200);
m_browser.SetHeight(800);
m_browser.Navigate("www.web.de", 0, 0, 0, 0);
```

Abbildungsverzeichnis

1.1	Gesamtüberblick der Systemarchitektur - 1.Version	6
2.1	Vergleich der Mobilfunkraten, von 1999 und 2003, in Europa	10
2.2	Physikalische Medien	12
2.3	Größenvergleich: Bluetooth Chip mit 50 Pfennig Stück [14]	15
2.4	Scatternet mit 2 Piconets im Verbund [14]	17
2.5	Verbindungsaufbau Bluetooth [14]	19
2.6	Bluetooth Specification Protocol Stack. [14]	21
2.7	Übersicht der Profile und deren Abhängigkeiten voneinander [13]	25
2.8	Grobe Ansicht des Bluetooth - Moduls	28
2.9	Radio - Komponente	28
3.1	Anwendungsfall PES	30
3.2	Anwendungsfall PDA	32
3.3	Vorauswahlfenster auf dem PDA	35
3.4	Hauptfenster auf den PDA	36
3.5	Die Diashow wird im Hauptfenster des PDA selektiert	36
3.6	Eine Folie wird im Hauptfenster des PDA ausgewählt	37
3.7	Diashowauswahl auf dem PDA	37
3.8	Anwendungsfall PAE	38

3.9	Gesamtüberblick der Systemarchitektur - 2.Version	39
4.1	Gesamtübersicht des Labors	40
4.2	Gesamtüberblick der Systemarchitektur - 3.Version	48
5.1	Komponentendiagramm	50
5.2	Folien- und Bluetoothgeräteauswahl	58
5.3	Verbindungsaufbau zum Bluetoothgerät und Streaming der Dateien	59
5.4	Folienanzeige auf der PAE	61
5.5	Diashow Auswahl	63
5.6	Diashow Hauptfenster	65
5.7	Klassendiagramm	67
5.8	Systemkomponente PSE	68
5.9	Vorauswahlfenster auf dem PDA	69
5.10	Hauptfenster auf dem PDA	70
5.11	Diashowauswahl auf dem PDA	71
5.12	Anzeige der Folien auf dem Beamer.	75
5.13	Klassendiagramm der PAE	77
5.14	Systemkomponente der PAE	77

Tabellenverzeichnis

2.1	Vergleich WLAN mit Bluetooth	14
2.2	Die Protokolle und Schichten des Bluetooth-Protokollstacks	22
2.3	Abkürzungen für die Radio - Komponente	28
4.1	Technische Daten Desktop PC	41
4.2	Technische Daten PDA: <i>iPAQ H3870</i>	43
4.3	Das BTAccess Paket	44
4.4	Technische Daten Notebook	45
4.5	Technische Daten Beamer	46
4.6	Embedded Visual C++ für Windows CE	47
A.1	Methoden der Klasse: <i>CBtStack</i>	91
A.2	Methoden der Klasse: <i>CBtDevice</i>	92

Literaturverzeichnis

- [1] Weiser, Mark: Ubiquitous Computing

<http://www.ubiq.com/hypertext/weiser/UbiHome.html>
Zugriffsdatum: 29.07.2002

- [2] Weiser, Mark: Ubiquitous Computing

<http://nano.xerox.com/hypertext/weiser/UbiHome.html>
Zugriffsdatum: 29.07.2002

- [3] VDE Verband der Elektrotechnik, Elektronik und Informationstechnik Pressemitteilung vom 13.03.2002 - VDE: Hohes Technologie- und Marktpotenzial für „Ubiquitous Computing“

<http://www.vde.com/VDE/de/presse/marktpotential.htm>
Zugriffsdatum: 05.08.2002

- [4] IZT IZT, SFZ, IAT - Entwicklung und zukünftige Bedeutung mobiler Multimediadienste - Werkstattbericht Nr.49

<http://www.izt.de/mmd/download.asp>
Zugriffsdatum: 06.08.2002

- [5] Andrew S. Tanenbaum, Maarten van Stehen: Distributed Systems: Principles and Paradigms, Prentice Hall

- [6] Karner, Harald, Entwicklungsumgebung für Windows CE, SEMINAR/PROJEKT INFORMATIONSSYSTEME WS2000/01, Version 1.1a , 22.1. 2001.

www.iicm.edu/research/seminars/ws_00/karner.pdf
Zugriffsdatum: 01.07.2003

- [7] Das Fünf-Sekunden-Programm , c't 2003, Heft 14, Seite 26

- [8] James F. Kurose und Keith W. Ross: „Computer Networking A Top-Down Approach Featuring the Internet“, Addison Wesley, 2nd Ed., 2002

- [9] Muller, Nathan J.: “Bluetooth Demystified“, The McGraw-Hill Companies, 2000

- [10] Senft, Matthias: AIS-Vortrag WS 2001/02 „Bluetooth“ 2002
<http://users.informatik.fh-hamburg.de/~ais/ws20012002/abstr/bluetooth.zip>
Letzter Zugriff: 01.07.2003
- [11] rfi mobile technologies AG: rfi WhitePaper „Bluetooth Technologie“
<http://www.rfi.de/downloads/blue.pdf>
Letzter Zugriff: 01.07.2003
- [12] SIG: „The Official Bluetooth Website“
<http://www.bluetooth.com/>
Letzter Zugriff: 01.07.2003
- [13] Bluetooth - Usage Models, Protocols, and Profiles
http://www.holtmann.org/lecture/bluetooth/BT_ProtocolsProfiles.ppt
Letzter Zugriff: 01.07.2003
- [14] IDG Interactive GmbH, tecChannel.de
<http://www.tecchannel.de>
Letzter Zugriff: 01.07.2003
- [15] Windows CE 3.0 Produktkatalog - Übersicht
<http://www.microsoft.com/germany/produkte/overview.asp?siteid=10772>
Letzter Zugriff: 01.07.2003
- [16] eMbedded Visual Tools 3.0 - 2002 Edition von Microsoft
<http://www.microsoft.com/downloads/details.asp>
Letzter Zugriff: 15.04.2003
- [17] TDK Bluetooth-USB-Adapter
<http://www.tdksys.com/download/pdf/deusbgerman.pdf>
Letzter Zugriff: 15.04.03
- [18] Beamer von Sharp PG-M25x WLAN-Fähig
<http://www.projektor-beamer.de/projektoren/sharp/pdf/pgm20-pgm25.pdf>
Letzter Zugriff: 29.06.03
- [19] Firma High Point Software
<http://www.high-point.com>
Zugriffsdatum: 06.08.2002
- [20] Nico Edtinger, „Bluetooth“
<http://www.chatternation.net/alpha/docs/bluetooth/>
Letzter Zugriff: 05.07.03
- [21] Interest, „Lexikon IT-Fachbegriffe“
<http://www.interest.de/cgi-bin/lexika/>
Letzter Zugriff: 05.07.03
- [22] Siemens, „Das Siemens Online Lexikon“
http://w3.siemens.de/solutionprovider/_online_lexikon/
Letzter Zugriff: 05.07.03

Versicherung über Selbstständigkeit

Hiermit versichern wir, daß wir die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfaßt und nur die angegebenen Hilfsmittel benutzt haben.

Ort, Datum

Unterschrift des Studenten

Ort, Datum

Unterschrift des Studenten