



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Diplomarbeit

Nutzung von RFIDs in einem auf
Web-Services basierenden Informationssystem

vorgelegt von

Klaus Stammermann

am 29. Juni 2005

Studiengang technische Informatik

Betreuender Prüfer: Prof. Dr.rer.nat. Gunter Klemke

Zweitgutachter: Prof. Dr. Kai von Luck

Fachbereich Elektrotechnik und Informatik
Department of Electrical Engineering and Computer Science

Nutzung von RFIDs in einem auf Web-Services basierenden Informationssystem

Stichworte RFID, Web-Services, Informationssystem, Bluetooth, WLAN, ubiquitäre Systeme

Zusammenfassung

Computer werden immer kleiner und leistungsfähiger. Daher kommen sie auch zunehmend in bisher unüblichen Umgebungen zum Einsatz. Diese Tatsache und die Verfügbarkeit von verschiedenen Funktechnologien wird in dieser Arbeit durch einen mobilen Assistenten für einen Museumsbesuch zur praktischen Anwendung gebracht. Dabei werden verschiedene Funktechniken kombiniert, um dadurch dem Nutzer eine komfortable Bedienung zu ermöglichen. Es wird ein Beispiel aufgezeigt, wie ubiquitäre Systeme auch in bisher informatikfremden Umgebungen hilfreich eingesetzt werden können.

Utilisation of RFIDs in an information system based on web services

Keywords RFID, web-service, information system, Bluetooth, WLAN, ubiquitous systems

Abstract

Computers become smaller and increasingly powerful. Therefore they are used in unusual environments. This fact and the availability of different radiotechnologies are coming to a practical use in this thesis by developing a mobile assistant for going to museum. In this project different radiocommunication technologies are combined into a comfortable assistant. It shows an example of using ubiquitous computing in a for computers unusual area.

Inhaltsverzeichnis

1. Einleitung	7
1.1. Aufbau der Arbeit	7
1.2. Motivation	8
1.3. Aktueller Stand	9
1.4. Aufgabenstellung	10
1.5. Zielsetzung	10
2. Grundlagen	12
2.1. RFID	12
RFID-Typen	13
Funktionsweise	14
Sicherheit und Datenschutz	14
2.2. Bluetooth	16
Geschichte	16
Verbindungsarten	16
Bluetooth Netzwerke	17
Fast Frequency Hopping Verfahren	19
Der Bluetooth-Protokollstack	19
Bluetooth Profile	21
2.3. WLAN	22
Standards	22
Topologien	23
Funktionsweise	24
2.4. Wechselwirkungen zwischen Bluetooth und WLAN	24
2.5. Java	25
J2ME	25
PersonalJava	26
2.6. Verteilte Systeme	26
Java RMI	27
CORBA	27
SOAP	28

2.7. Zusammenfassung	28
3. Anforderungen	29
3.1. Bereich der Informatik	29
3.2. Allgemeine Anforderungen	30
3.3. Lösungsansätze	31
Anpassung der Informationstafeln	31
Dynamische Informationstafeln	32
Mobile Informationstafeln	32
3.4. Anforderungen an das Lösungssystem	34
Gesamtsystem	34
Mobiler Client	35
Objektidentifikation	35
Server	36
3.5. Zusammenfassung	36
4. Design	37
4.1. Gesamtsystem	37
4.2. Clientsoftware	39
Informationsdarstellung	40
Lesegerätidentifikation	40
4.3. Serversoftware	41
Web-Service	42
Datenbank	43
4.4. Kommunikation	43
PDA - RFID-Lesegerät	45
Client - Server	45
4.5. Zusammenfassung	45
5. Realisierung	46
5.1. Hardwarekomponente	46
PDA	47
Identifikation	48
5.2. Programmiersprache	50
5.3. Clientsoftware	50
Informationsdarstellung	51
Lesegerätidentifikation	52
Kommandosätze	53
5.4. Serversoftware	55
Web-Service	55
Datenbank	56

5.5. Kommunikation	57
PDA - Lesegerät	57
PDA - Server	58
5.6. Zusammenfassung	58
6. Systemtest	59
6.1. Hardware	59
6.2. Kommunikation	60
6.3. MuseumAssistant	62
Systemstart	62
Abrufen allgemeiner Informationen	63
Abrufen erweiterter Informationen	63
Nutzung unbekannter ID	65
6.4. Entwicklungsstand	66
Gesamtsystem	66
Clientsoftware	67
Serversoftware	67
6.5. Zusammenfassung	67
7. Fazit und Ausblick	69
7.1. Fazit	69
7.2. Ausblick	70
Multimedia	70
RFID-Leser	70
Merkliste	71
Nutzungsstatistiken	71
PC-Cafe	72
Mobile Datenbank	72
J2ME-Client	73
Einkaufsbegleiter	73
Tiererkennung	73
Glossar	75
Literaturverzeichnis	79
A. DTD Kommandosatz	82
B. DTD Objectlist	83
C. CD-Inhalt	84

Abbildungsverzeichnis

2.1. RFID-TAG	
http://www.computerworld.com/softwaretopics/erp/story/0,10801,84004,00.html	12
2.2. Energieversorgung passiver Transponder	
RFID-Handbook Kap.3 S.42	14
2.3. Bluetooth-Netzwerktopologien	
http://www.palowireless.com/infotooth/tutorial/baseband.asp	18
2.4. Bluetooth Protokoll Stack	19
2.5. Bluetooth-Profile	
http://www.microvision.co.kr/bluetooth/lecture/lecture_protocol_2.htm	22
3.1. Systemaufbau	33
3.2. Use Case: Grund Informationen	34
3.3. Use Case: Erweiterte Informationen	35
4.1. Use Case Diagramm	38
4.2. Klassendiagramm: Clientsoftware	39
4.3. Zustandsdiagramm: Identifikationsdiagramm	41
4.4. Klassendiagramm: Serversoftware	42
4.5. Sequenzdiagramm: Abfrage Grundinformationen	44
4.6. Sequenzdiagramm: Abfrage erweiterte Informationen	44
5.1. iPAQ H5550	47
5.2. SONMicro RFID-Reader	48
5.3. Bluetoothmodul von PicoPlug	49
5.4. MuseumAssistant	51
5.5. Beispiel Kommunikationsablauf	55
6.1. MuseumAssistant	63
6.2. Anzeige allgemeiner Informationen	64
6.3. Anzeige erweiterter Informationen	64
6.4. Anzeige bei unbekannter ID	65

1. Einleitung

Durch die ständige Entwicklung sowohl in der Informatik als auch in der Computertechnologie ist es möglich, die Informatik in immer neue Bereiche des Alltags zu integrieren. Neben den stationären Systemen, die nur lokal einen Computer nutzen, gibt es verteilte Systeme. Bei diesen werden einzelne Aufgaben von unterschiedlichen Systemen erledigt. Das Zusammenspiel dieser bietet dann ein effektiveres und flexibleres Gesamtsystem, als es mit einer stationären Einzelanwendung möglich wäre.

Ein weiteres Gebiet der Informatik beschäftigt sich damit, Systeme für den Alltag sinnvoll nutzbar zu machen. Man spricht hierbei von ubiquitären Systemen¹. Diese sollen hauptsächlich dem Nutzer eine einfache Bedienung bieten. Zudem ist diesem nicht einmal bekannt, dass er ein Computersystem bedient bzw. was alles außer dem eigentlichen Bedienelement zum genutzten System gehört. Das ermöglicht ein wesentlich breiteres Feld von Anwendern, da auch Menschen mit einer ablehnenden Haltung gegenüber technischen Systemen als Nutzer in Frage kommen.

Genau deswegen soll in dieser Arbeit ein mobiles Informationssystem entwickelt werden, welches die beiden oben genannten Bereiche der Informatik verbindet. Es soll ein effektives, leicht zu bedienendes und vor allem auch flexibles System entstehen, welches dem Anwender eine einfache Bedienung bietet. Zudem soll für den Nutzer nicht erkennbar sein, welchen Umfang das Gesamtsystem hat. Diese Details sind zum Einen unwichtig für den Nutzer, da diesen nur die Anwendung und den daraus entstehenden Nutzen interessiert. Zum Anderen dürften die Details Menschen abschrecken, da die Beschreibung des System komplizierter ist als die eigentliche Anwendung.

1.1. Aufbau der Arbeit

Die Arbeit ist in sieben Kapitel unterteilt, welche die Struktur und den Entwicklungsablauf dieser Diplomarbeit wiedergeben.

¹Als „ubiquitäre Systeme“ bezeichnet man Computersysteme, die dem Nutzer insgesamt oder auch nur teilweise als solches nicht erkennbar sind.

Wir befinden uns im ersten Kapitel, welches die Einleitung umfasst. Die folgenden Teile dieses Kapitels sollen den Leser an das Thema heranzuführen. Zudem wird hier neben der Motivation zu diesem Thema auch der aktuelle Stand der angewandten Lösungen beschrieben. Es folgt eine Beschreibung der Aufgabenstellung sowie die Erläuterung, welches Ziel diese Arbeit verfolgt.

Im nächsten Kapitel werden die unterschiedlichen Grundlagen zusammengefasst. Dabei werden grundlegende Teile der genutzten Techniken erläutert, um dem Leser das nötige Basiswissen zu vermitteln. Es soll das Verständnis für den Einsatz der Techniken erläutert werden, was sowohl Hardwarekomponenten sowie Softwaretechniken beinhaltet.

Das dritte Kapitel umfasst die Ergebnisse der Anforderungsanalyse. Hier wird aufgezeigt, welche Funktionen das System bieten soll und worauf dabei die Schwerpunkte gelegt werden. Es werden verschiedene Ansätze aufgezeigt und die gewählte Variante begründet.

Das vierte Kapitel legt dem Leser das Design der entwickelten Software nahe. Darin werden die verschiedenen Möglichkeiten für den Anwender aufgeschlüsselt und durch entsprechende Diagramme dargestellt.

Das folgende Kapitel befasst sich mit der Realisierung der Software. Die verschiedenen genutzten Softwarebibliotheken werden dargestellt und deren Nutzungsgründe erläutert. Zudem wird dem Leser das Zusammenspiel der unterschiedlichen Komponenten der Software beschrieben, so dass der interne Ablauf deutlich wird.

Dem sechsten Kapitel sind die Systemtests zu entnehmen. Hier werden die Ergebnisse der Tests und der aktuelle Stand des entwickelten Systems beschrieben.

Das abschließende siebte Kapitel beinhaltet das Fazit sowie einen Ausblick. Hier wird sich kritisch mit der Arbeit auseinander gesetzt und die unterschiedlichen Schritte werden bewertet. Zudem werden verschiedene Erweiterungs- sowie Einsatzmöglichkeiten dargestellt, welche sinnvoll zu realisieren wären, aber durch den begrenzten Zeitrahmen nicht Teil dieser Arbeit geworden sind.

1.2. Motivation

Um einen Museumsbesuch interessanter zu machen, wird im Laufe dieser Arbeit ein unterstützendes Informationssystem entwickelt. Durch die Nutzung aktuell technischer Möglichkeiten soll den Besuchern ein Informationsdienst bereitgestellt werden, welcher individuell über einen PDA genutzt werden kann.

Dafür lässt sich die Motivation zu dieser Arbeit in zwei Bereiche aufteilen. Zum Einen soll dem Nutzer ein komplexes System durch eine möglichst einfach Bedienung bereitstellen

werden. Dieser Aspekt ist sehr wichtig für ein System, welches an die breite Masse gerichtet ist. Sobald die Bedienung aufwendig oder umständlich wird, schreckt dieses viele mögliche Nutzer ab, was unbedingt verhindern werden soll. Durch den Einsatz des Systems durch Menschen aus den verschiedensten Sozial- und Bildungsbereichen wird die ablehnende Haltung gegenüber neuer Technologie abgebaut, wie sie z.B. auch bei vielen älteren Menschen zu finden ist.

Der andere Schwerpunkt der Motivation liegt in der Entwicklung eines flexiblen Systems. Durch einfache und zum Teil minimale Anpassungen soll das entwickelte System an die unterschiedlichsten Umgebungsvorgaben anpassbar werden. Dadurch werden sehr viele verschiedene Einsatzgebiete möglich. Zudem sollen verschiedene Funktechnologien in Kombination zum Einsatz kommen, so dass sowohl Flexibilität als auch Mobilität erreicht wird.

Als Nebenbereich der direkten Motivation soll auch auf die Nutzung von RFID hinweisen werden. Diese Technologie ist viel diskutiert und besitzt keinen guten Ruf in der Allgemeinheit. Besonders Datenschützer tragen zu dieser Meinung bei. In dieser Arbeit wird diese Technik in einem Bereich eingesetzt, der im Hinblick auf den Datenschutz unbedenklich ist. Dadurch soll ein möglicher Nutzen aufgezeigt werden, durch den evtl. das allgemeine Bild von RFIDs verbessert wird.

1.3. Aktueller Stand

In den meisten Museen werden bisher eher Lösungen genutzt, die technisch einfach gestaltet sind. Neben den klassischen Führungen, bei denen ein Mitarbeiter die Besucher durch die Ausstellung begleitet und dabei Informationen verbal wiedergibt, bieten manche Museen inzwischen auch technische Unterstützung an.

Zum Einen werden die Führungen durch Funkkopfhörer unterstützt, so dass alle Teilnehmer den Mitarbeiter deutlich verstehen können. Teilweise ist sogar eine Sprachauswahl möglich, wobei hierbei nur vorbereitete Sprachdateien abgespielt werden. Somit sind dabei keine individuellen Nachfragen möglich.

Des Weiteren bieten viele Museen Onlineportale an, um sich bereits vor dem eigentlichen Museumsbesuch Informieren zu können. Trotz der teilweise eingesetzten Webcams ist es für den Besucher allerdings schwierig, den Zusammenhang zwischen der Onlinerecherche und dem eigentlichen Museumsbesuch herzustellen, was besonders durch den zeitlichen Unterschied deutlich wird.

Abschließend sind noch Video- bzw. Tonmaterial zu nennen. Manche Museen bieten Filmvorführungen an, bei denen der Besucher sich in einen entsprechenden Vorführraum begeben muss. Dadurch entsteht natürlich eine räumliche Trennung. Zudem sind dabei indi-

viduelle Informationsabfragen nur unter Umständen möglich. Audioinformationen hingegen werden teilweise stationär bei den Ausstellungsobjekten eingesetzt. Dabei kann der Besucher z.B. durch Betätigen eines Schalters sich Informationen vorspielen lassen. Auch hierbei wird allerdings eine Vorauswahl der Informationen getroffen.

1.4. Aufgabenstellung

Ein Museum ist allgemein eher ein Bereich, in dem nicht mit neuer Technologie gerechnet wird. Hier können sich Besucher über unterschiedliche Themenbereiche informieren. Um sowohl verschiedenste Menschen als Nutzer zu erreichen und eine komplexes System mit einfacher Bedienung anzubieten, soll in dieser Diplomarbeit ein Informationssystem für Museen entwickelt werden.

In diesem Szenario ist es möglich, dass sich ein Besucher vor Betreten der Ausstellung einen PDA mit einer Informationssoftware im Museum ausleiht. Diesen PDA führt er während seines Besuchs mit sich und kann bei Interesse Informationen zu den einzelnen Ausstellungsstücken abrufen. Somit ist es ihm möglich, sich individuell zu informieren, was den Besuch wesentlich interessanter macht.

Um die Bedienung möglichst einfach zu halten, sollen die Ausstellungsstücke berührungslos identifiziert werden. Dazu bekommt man zu dem PDA ein entsprechendes Lesegerät bereitgestellt. Dieses wird per Bluetooth vom PDA gesteuert, so dass der Nutzer das Lesegerät nur in die Nähe des Objekts bzw. dessen RFID-Tags halten muss.

Des Weiteren soll eine zentrale Datenverwaltung des System ermöglicht werden. Dazu kommt ein Server zum Einsatz, welcher per WLAN den einzelnen PDAs Informationen bereitstellt. So können die Daten einfach gewartet werden und das System bleibt für das Personal überschaubar. Für den Nutzer ist es aber unwichtig, dass ein Server genutzt wird, da die entsprechenden Abfragen vom System automatisch erledigt werden und die Kommunikation im Hintergrund abläuft.

1.5. Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines entsprechenden Grundgerüsts für ein mobiles, verteiltes Informationssystem. Dazu soll sowohl die Clientsoftware als auch der Server exemplarisch realisiert werden, so dass ein Prototyp des Gesamtsystems am Ende der Arbeit verfügbar ist. Dabei soll zudem darauf geachtet werden, dass alle Komponenten des Systems einfach austauschbar werden. So soll eine flexible Anpassung an das gegebene Umfeld ermöglicht werden.

Des Weiteren soll die Kommunikation sowohl zwischen PDA und Lesegerät als auch zwischen PDA und Server automatisiert ablaufen. Durch diese Automatisierung ist eine Nutzung im Hintergrund möglich, was die Bedienung des System enorm erleichtert.

Für die unterschiedlichen Kommunikationsarten im entwickelten System sind Mechanismen wichtig, die flexible Anpassung und automatisierte Kommunikationen unterstützen. Da das Lesegerät aber direkt als Hardware angesprochen wird und somit keine erweiterten Softwareschnittstellen bietet, muss hierfür ein Mechanismus entwickelt werden, der automatisch die Hardware erkennt und die dafür benötigten Steuerungskommandos dem System bereitstellt.

2. Grundlagen

Im folgenden Kapitel werden die verschiedenen genutzten Technologien vorgestellt. Hierzu enthält diese Arbeit zu jeder Technologie einen allgemeinen Überblick, wobei auf die genutzten Teile etwas genauer eingegangen wird. Durch dieses Kapitel werden die Grundlagen für die Ausarbeitung des Systems geschaffen. Zuerst wird die Hardwareseite betrachtet, welche RFID , Bluetooth sowie WLAN und deren Zusammenarbeit bzw. parallelen Betrieb beinhaltet. Darauf folgt dann das nötige Basiswissen für die Softwareentwicklung sowie für die Entwicklung der Systemstruktur.

2.1. RFID

RFIDs sind eine viel diskutierte Technologie, die in verschiedenen Industriezweigen genutzt wird bzw. vor der Einführung steht (siehe Böhmer u. a. (2005)). Hierbei handelt es sich um eine Art Barcodenachfolger, welche kontaktlos per Funk ausgelesen werden kann. Allerdings bietet diese Technologie mehr als nur eine Identifikationsmöglichkeit, wie es durch diesen Abschnitt deutlich werden soll. Des Weiteren sind RFID-Tags¹ sehr günstig produzierbar (ab ca. 0,05 Euro) und bieten einen Datenspeicher, der derzeit bis zu einem Megabyte umfassen kann. Somit bieten sich eine Vielzahl von verschiedenen Einsatzmöglichkeiten, was auch durch die Nutzung in dieser Arbeit gezeigt wird (siehe Stein (2005)).

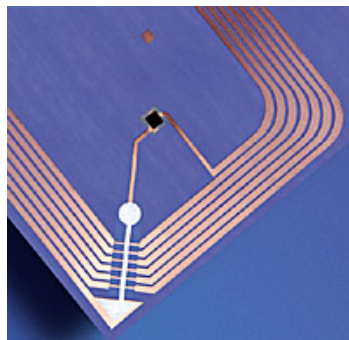


Abbildung 2.1.: RFID-Tag

¹Als Tag bezeichnet man eine Funkmarkierung in der RFID-Technologie.

Im Folgenden soll ein Basiswissen vermittelt werden, so dass der Leser diese Technik und deren Nutzung versteht. Dafür werden sowohl die Funktionsweise wie auch die unterschiedlichen Typen beschrieben. Da Datensicherheit und Datenschutz eine wichtige Komponente in der Diskussion um RFIDs ist, wird darauf auch noch kurz eingegangen. Für tiefergehende Informationen wird auf entsprechende Literatur verwiesen (z.B. Finkenzeller (2003)).

RFID-Typen

Technisch gesehen ist ein RFID-Tag ein Transponder, wobei diese Bezeichnung die Kombination von einem Transmitter und einem Responder aufzeigt. Somit bieten Tags eine bidirektionale Kommunikation. Dabei gibt es zwei verschiedene RFID-Typen, die sich durch die Energieversorgung unterscheiden. Aktive Transponder haben eine eigene Stromversorgung, welche ihnen eine höhere Reichweite ermöglicht. Dafür sind sie teurer in der Produktion. Passive Transponder hingegen erhalten ihre Energie durch den Leser. Somit werden sie nicht durch begrenzte Akkuleistungen in der Nutzung eingeschränkt und können flexibel auch über längere Zeiträume eingesetzt werden. Zwar bieten sie eine geringere Reichweite, aber durch die geringen Produktionskosten sind sie für wesentlich mehr Gebiete in der Industrie interessant.

Eine weitere Unterteilung lässt sich anhand der genutzt Frequenz und der damit verbundenen Reichweite treffen. Niedrigfrequente Tags nutzen einen Frequenzbereich von ca. 30-500kHz² und erreichen damit eine geringe Reichweite von nur wenigen cm. Allerdings bieten diese die geringsten Produktionskosten, so dass ihre Verwendung relativ häufig ist (z.B. Zugangskontrollen und Lagerverwaltung). Im mittleren Frequenzbereich von ca. 10-15MHz³ sind die so genannten SmartLabels anzutreffen. Mit mittlerer Reichweite sowie mittleren Kosten bieten sie für verschiedenste Bereiche gute Einsatzmöglichkeiten. Hochfrequente RFIDs nutzen verschiedene Frequenzbereiche⁴ und erreichen damit eine Reichweite von bis zu 30m. Zwar sind sie die teuerste Variante, aber durch die enorme Reichweite sind sie für viele Bereiche (wie z.B. Mautsysteme oder Güterwagenerkennung) interessant.

Abschließend ist zu den unterschiedlichen Typen zu sagen, dass hohe Reichweiten, hohe Frequenzen und Erweiterungen wie Kryptographiemodule sehr viel Energie benötigen. Da Energien für diese Funktionen schwierig und teilweise auch gar nicht durch Induktion in diesem Bereich erreichbar sind, werden die Funktionalitäten oft nur bei aktiven Transpondern angeboten. Allerdings sind aktive Transponder von einer Stromversorgung abhängig, so dass genau untersucht werden sollte, ob entsprechende Reichweiten und Module zwingend benötigt werden.

²häufig 125kHz

³häufig 13,56MHz

⁴850 - 950 MHz, 2,4 - 2,5 GHz, 5,8 GHz

Funktionsweise

Sobald ein Lesegerät in Reichweite eines RFID-Tags ist und ein entsprechendes Lesesignal sendet, reagieren die Tags. Aktive Transponder können direkt mit der Kommunikation beginnen, so dass es sich hierbei um eine relativ unproblematische Kommunikation handelt. Anders ist es allerdings bei den passiven Tags. Diese müssen zuerst genügend Energie gewinnen, um eigene Signale zurücksenden zu können. Indem das Lesegerät die Antenne des Tags in Schwingung bringt, gewinnt dieses aus der entstehenden Induktion eigene Energie. Zur Veranschaulichung wird auf Abbildung 2.2 hingewiesen.

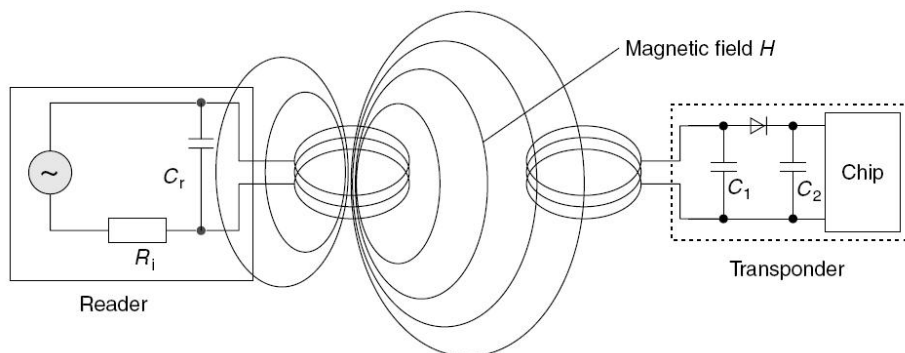


Abbildung 2.2.: Energieversorgung passiver Transponder

Um die Kommunikation problemlos ablaufen zu lassen, bestehen feste Sendereihenfolgen, so dass sich Leser und Transponder nicht gegenseitig stören. Hierbei ist zu beachten, dass bei passiven Tags die Sendedauer des Lesers wesentlich höher ist, um benötigte Induktionen für die Energieversorgung aufbauen zu können. Im Falle von mehreren Tags in Reichweite des Lesers wird nach dem so genannten Aloha-Protokoll verfahren, so dass alle entsprechenden Tags nacheinander angesprochen werden. Für den genauen Ablauf wird dem Leser entsprechende Literatur nahe gelegt (z.B. Finkenzeller (2003) oder Burdet (2004)).

Sicherheit und Datenschutz

Da es sich bei RFIDs um eine kontaktlose Kommunikationstechnik handelt, darf die Sicherheit nicht vernachlässigt werden, da Daten unbemerkt ausgelesen werden können. Das kann zu Sicherheits- und Datenschutzproblemen führen. Hinzu kommt die Klartextübertragung, die bei den meisten RFID-Typen Standard ist. Aktive Transponder sind daher inzwischen auch mit internen Kryptographiemodulen erhältlich. Dadurch lässt sich eine Verschlüsselung erreichen, so dass eine höhere Sicherheit gegeben ist.

Passive Transponder bieten diese Möglichkeit leider nicht, da für die Kryptographie zu viel Energie benötigt wird. Einzig die häufig geringe Reichweite bietet eine Art „Pseudosicherheit“. Daher sollte diese Technik möglichst in sicherheitsunkritischen Bereichen eingesetzt werden. Für tiefergehende Informationen wird auf entsprechende Literatur hingewiesen (z.B. Auf dem Hövel (2004)).

Da Sicherheit im engen Zusammenhang mit Datenschutz steht, muss auch dieser Bereich beachtet werden. Viele Datenschützer befürchten einen „gläsernen Kunden“, da viele Daten über z.B. Kaufgewohnheiten unbemerkt mit RFIDs ermittelt werden können. Zwar ist die Datenerhebung allgemein gesetzlich geregelt (siehe auch Bundesverband Informationswirtschaft (2004)), aber durch die Möglichkeit des unbemerkten Auslesens werden an vielen Stellen Bedenken geäußert.

Zum Einen möchte der Autor darauf hinweisen, dass viele Kunden für Rabatt durch eine Kundenkarte ähnliche Informationen freiwillig den Unternehmen geben oder sich für einen VIP-Zugang sogar entsprechende Chips implantieren lassen⁵. Zum Anderen würde eine Ausweitung der Problematik an dieser Stelle den Rahmen der Arbeit sprengen. Daher soll dieser Abschnitt nur darauf hinweisen, dass es viele Diskussion um dieses Thema gibt und sich daher interessierte Leser an anderer Stelle entsprechende Informationen besorgen möchten (z.B. Ziegler (2005), Löding (2005) oder Böhmer u. a. (2005)).

Zwar ist sowohl Datensicherheit als auch Datenschutz wichtig, aber im Rahmen der Nutzung von RFIDs in dieser Arbeit eher nebensächlich. Hier werden keine sicherheitsrelevanten Daten sondern nur Erkennungsnummer der Ausstellungsobjekte ausgelesen. Zwar könnten Besucherprofile erstellt werden, aber es werden keine Daten in einem direkten Zusammenhang zum Nutzer gebracht, so dass auch der Datenschutz hier eine untergeordnete Rolle spielt. Erst bei einigen Erweiterungsmöglichkeiten, die in Kapitel 7.2 beschrieben werden, muss sowohl auf Datenschutz als auch auf Datensicherheit geachtet werden.

⁵Dies ist inzwischen z.B. im Club Baja Beach in Barcelona für den VIP-Bereich ein Muss (siehe z.B. Jones (2004))

2.2. Bluetooth

Bluetooth⁶ ist eine Funktechnik für den Nahbereich, die ursprünglich als Alternative zum Kabeldurcheinander zwischen Handys und deren Zusatzgeräten entwickelt worden ist. Inzwischen liegt das Einsatzgebiet sowohl bei verschiedener mobiler und als auch bei stationärer Hardware. So bieten z.B. viele aktuelle Handys und PDAs diese Technologie für den einfachen Datenaustausch, aber auch stationäre Systeme stellen Bluetooth zur Verfügung. Sobald ein Produkt das nebenstehende Logo aufweist, wird deutlich, dass dabei Bluetooth entsprechend der Spezifikation eingesetzt wird.



Mit den neuesten Chipsätzen lassen sich Entfernungen von über 200m überbrücken (siehe Zivadinovic (2004)). Durch diese Reichweite ermöglicht Bluetooth wesentlich mehr Einsatzgebiete als ein reiner Kabelersatz. Zur Kommunikation zwischen den einzelnen Geräten (z.B. zwischen PC und Drucker) müssen alle ein Bluetoothmodul haben. Da aber die kostengünstige Produktion eines der Hauptpunkte in der Entwicklung ist, sind diese Module weit verbreitet und die Technologie findet in immer mehr Bereichen ihre Verwendung. Zudem handelt es sich um eine lizenzfreie Entwicklung, wodurch die Verbreitung weiter unterstützt wird.

Geschichte

1994 begann bei Ericsson die Entwicklung von Bluetooth. Zusammen mit Intel, IBM, Toshiba und Nokia gründeten sie 1998 die Bluetooth SIG⁷, um Bluetooth als Quasi-Standard durchzusetzen. Bis April 2000 haben sich schon mehr als 2000 Firmen dieser Gruppe angeschlossen, woraus sich das Interesse der Industrie an einer solchen Technologie klar ablesen lässt. Ende 2004 wurde die Version 2.0 der Core Engine verabschiedet, mit der eine Geschwindigkeit von bis zu 3 Mbit/s erreichbar sind.

Verbindungsarten

Die Bluetoothspezifikation definiert zwei Kommunikationsarten, die für verschiedene Anwendungen genutzt werden können. Sowohl SCO (synchronous connection oriented) als auch ACL (asynchronous connectionless) werden im Folgenden erklärt.

⁶Der Name Bluetooth stammt von einem dänische König namens Harald I. Blauzahn Gormson. Dieser hat im Mittelalter verschiedene Bereiche Skandinaviens unter seiner Herrschaft vereint. Da Bluetooth auch verschiedene Gebiete über eine einheitliche Technologie vereinen soll, liegt die Namensableitung nahe. Daher stammt also der Name für diese Funktechnologie.

⁷<http://www.bluetooth.com>

SCO

Eine synchrone, verbindungsorientierte Verbindung wird typischerweise für Sprachverbindungen genutzt. Es handelt sich dabei um eine feste Punkt-zu-Punkt-Verbindung, die auch dann aufrecht erhalten wird, wenn keine Daten versendet werden. Pro Kanal werden 64 kB/s bereitgestellt, die der Applikation bis zum Verbindungsabbau zur Verfügung stehen. Von diesen Kanälen werden üblicherweise drei für eine Sprachverbindung genutzt, um Quality of Service zu sichern.

ACL

Asynchrone, verbindungslose Übertragungen werden in zwei Gruppen unterteilt. Es gibt eine symmetrische Übertragung, bei der sowohl in Sende- wie auch in Empfangsrichtung die gleich Bandbreite zur Verfügung steht. Im Gegensatz dazu steht die asymmetrische Kommunikation, die unterschiedliche Bandbreiten für die jeweilige Richtung bereitstellt.

Bei beiden werden paketvermittelte Punkt-zu-Mehrpunkt-Verbindungen genutzt, die nie dauerhaft sind. Sie werden nur während der Übertragungsdauer erstellt, umso nicht unnötig Ressourcen zu belegen. Zudem können Pico-Netze aufgebaut werden, in denen adressierte Datenpakete direkt an das Zielgerät übertragen werden.

Bluetooth Netzwerke

Da über Bluetooth eine Übertragung zu allen entsprechenden Geräten in Umkreis der Sendeleistung realisierbar ist, sind somit auch Netzwerke mit diesen Geräten möglich. Daher ist dieses in der Spezifikation berücksichtigt worden, so dass das gesamte System flexibel einsetzbar ist.

Adhoc-Netzwerke

Bluetooth bietet keine feste Infrastruktur, da durch die Funkverbindungen das System flexibel wird. Daher handelt es sich immer um Adhoc-Netzwerke, welche sich im laufenden Betrieb verschiedene Veränderungen erfahren können. Wenn ein Gerät neu in Reichweite erkannt wird oder eines entfernt wird, muss das Netzwerk darauf entsprechend reagieren.

Sobald ein neues Gerät in Reichweite kommt, kann zu diesem eine neue Verbindung erstellt werden. Zudem besteht auch die Möglichkeit, dass Geräte im laufenden System das Netz verlassen, so dass sich dadurch die Rahmenbedingungen verändern. Um das Netz so flexibel wie möglich zu halten, ist es möglich, dass Geräte als Router arbeiten. Die Auswahl des

Gerätes, welches als Router fungieren soll, wird in der Spezifikation beschrieben. Dadurch können Geräte, die nicht in direkter Reichweite zueinander sind, über weitere miteinander kommunizieren.

Topologien

Da durch diese Rahmenbedingungen das Netzwerk sehr flexibel ist, gibt es verschiedene Topologien. In Abbildung 2.3 sind verschiedene Topologiearten zu erkennen. Dabei zeigt a eine einfache Verbindung, b ein Pico-Netz und c stellt mehrere kombinierte Pico-Netze dar, die zusammen ein Scatter-Netz bilden.

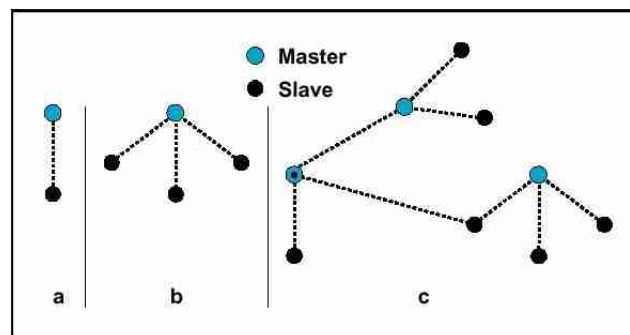


Abbildung 2.3.: Bluetooth-Netzwerktopologien

In Pico-Netzen gibt es immer nur einen Master und mindestens einen Slave. Bis zu acht Geräte können ein solches Netz bilden. Dabei gibt es eine eindeutige Aufgabenverteilung, wobei der Master die Kommunikation steuert. Sobald aber im Netzwerk nicht jeder Slave den Master erreichen kann, werden weitere Geräte zu einem Master. Diese Kombination von mehreren Pico-Netzen nennt man Scatter-Netze.

Hierbei hat jedes Gerät einen Master bzw. ist es selber. Manche Geräte sind für andere ein Master und stehen selbst zu anderen als Slave zur Verfügung. Dadurch werden Master-Master-Verbindungen verhindert, so dass eine Kommunikation immer definiert ist. Somit lassen sich Scatter-Netze in mehrere Pico-Netze unterteilen, wobei einige Teilnehmer in mehreren wieder zu finden sind. Zudem leitet jeder Master für seine Slaves das Paging-Verfahren ein und erstellt entsprechende Verbindungen.

Fast Frequency Hopping Verfahren

Bei den Funkverbindungen zwischen Bluetoothgeräten wird das Fast Frequency Hopping Verfahren eingesetzt, um eine relativ hohe Unempfindlichkeit gegenüber Störungen zu erreichen. Das komplette Frequenzband wird in 79 Kanäle⁸ unterteilt, zwischen denen 1600 mal pro Sekunde gewechselt wird. Falls es auf einer Frequenz eine Störung gibt, wird somit die Verbindung für nur 625 μsec unterbrochen. Die dort verlorenen Daten können später wiederholt gesendet werden.

Außerdem soll das Fast Frequency Hopping Verfahren eine höhere Sicherheit für die Verbindung bieten. Dieses ist aber nur bedingt wahr, da, sobald die Wechselreihenfolge der Frequenzen bekannt ist, die Daten direkt abgehört werden können. Es findet hier also keine Verschlüsselung statt, so dass von Abhörsicherheit nicht gesprochen werden kann.

Der Bluetooth-Protokollstack

Der Bluetooth-Protokollstack bietet verschiedene Protokolle, die aufeinander aufbauen. Als Übersicht bietet Abbildung 2.4 einen ersten Einblick.

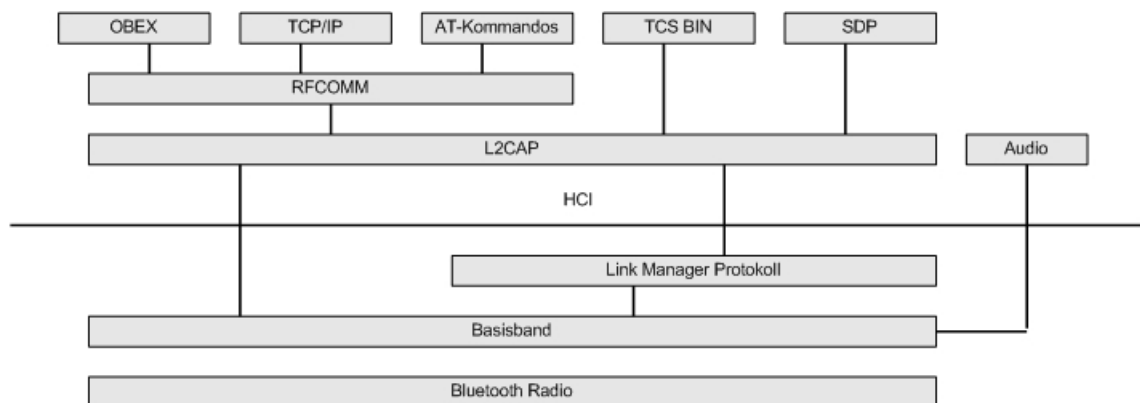


Abbildung 2.4.: Bluetooth Protokoll Stack

Da für diese Diplomarbeit nur das Protokoll RFCOMM genutzt wird, werden die anderen nur kurz erklärt. Dadurch wird ein Überblick deutlich gemacht, den der Leser bei Interesse selbst vertiefen kann⁹. Dafür werden die allgemeinen Zusammenhänge der Protokolle beschrieben, so dass dem Leser ein Überblick deutlich wird.

⁸In Frankreich in 27 Kanäle, da dort ein kleineres Band zur Verfügung steht.

⁹siehe hierzu z.B. Wollert (2002) oder auch Kirst (200)

Bluetooth Radio liegt allen Protokollen zugrunde. Hier wird die interne Hardwareansteuerung der Funkverbindung durchgeführt. Die Steuerung dieser Einheit erfolgt durch das Basisband. Neben der Verwaltung der Datenkanäle und des Frequenzhoppings wird auch hier der Aufbau von Piconetzen gesteuert. Zudem werden im Basisband Pakete erstellt und die Adressierung von Bluetooth-Geräten übernommen. Somit bilden Bluetooth Radio und das Basisband die grundlegenden Steuereinheit der Funkverbindung.

Auf diesen beiden setzt sowohl das Link Manager Protokoll¹⁰ sowie Audioverbindungen auf. LMP bietet das Verbindungsmanagement. Neben verschiedenen Sicherheitselementen¹¹ werden hier Verbindungen zwischen den Geräten auf Controllerebene verwaltet. Mit den unterliegenden Protokollen bietet HCI ein Programmierinterface. Dadurch sind verschiedene Befehle und Statusregister erreichbar, so dass eine Programmierung vereinfacht wird.

Die Steuerung der Audioverbindungen setzt auch auf dem Basisband auf, da diese relativ leicht zu realisieren sind. Audioverbindungen versenden sowohl über SCO sowie über ACL Daten. SCO wird für Sprachdatenverbindungen und das Zweite für Audiodatenübertragung wie z.B. MP3- und WAV-Dateien verwendet. Das immer beliebter werdende VoIP nutzt übrigens ACL-Verbindungen.

In der nächsten Protokollschicht finden wir L2CAP. Dieses übernimmt das Verbindungsmanagement für Datenkanäle. Somit finden hier die höher liegenden Protokolle ihre Datenkanäle, um mit der entsprechenden Gegenstelle zu kommunizieren. Dabei ermöglicht es verbindungsorientierte und verbindungslose Verbindungen und Multiplexing. Zudem verhindert es, dass höhere Protokollschichten Pakete über 64kB Größe verschicken.

Für die Erkennung der verschiedenen Dienste wird das SDP genutzt. Es steuert zu den angebotenen Diensten die Absprache zwischen den Geräten, so dass erkannt wird, ob eine Kommunikation zwischen verschiedenen Protokollen möglich ist. Es bildet mit dem Basisband, LMP und L2CAP die Bluetooth Kernprotokolle.

RFCOMM wird auch Kabelersatzprotokoll genannt, was die Funktion auch sehr gut beschreibt. Hiermit ist es möglich bis zu 60 virtuelle serielle Schnittstellen nach dem RS232-Standard zu simulieren. Damit sind viele Anwendungen, die über eine serielle Schnittstelle mit Modulen kommunizieren, nun auch mobil einsetzbar. Dazu müssen die entsprechenden Module nur über eine Bluetooth-Schnittstelle erweitert werden, wie es auch in dieser Diplomarbeit geschieht.

Für Telekommunikation wird TCS BIN eingesetzt. Dieses ist ein bit-orientiertes Protokoll, welche die Steuerung von Telefonverbindungen übernimmt. Dabei enthalten sind Rufkontrolle,

¹⁰Dieses Protokoll wird häufig mit LMP oder LM (für Link Manager) abgekürzt.

¹¹Authentisierung, Verschlüsselung, Keygeneration

Verbindungsaufbau, Sprachübertragung und Datenübertragung. Da aber auch Telekommunikationsverbindungen über AT-Kommandos gesteuert werden, wird dieses Steuerungsprotokoll unterteilt. Diese Kommando-Verbindung setzt auf dem eben beschriebenen RFCOMM auf.

Des Weiteren findet sich oberhalb von RFCOMM TCP/IP. Dieses Protokoll ist sehr weit verbreitet und bietet die Grundlage für verschiedenste Verbindungen. Es wird sowohl in lokalen Netzwerken sowie auch für verschiedene Internetverbindungen eingesetzt. Somit bietet Bluetooth mit der Bereitstellung dieses Protokolls eine breites Einsatzgebiet.

Abschließend ist noch OBEX zu nennen. Dieses kommt aus dem Bereich der Infrarot-Schnittstelle und soll einen weitgehend geräteunabhängigen, spontanen Datenaustausch ermöglichen. Aufgrund der vielfältigen Einsatzmöglichkeiten hat sich OBEX inzwischen zu einem unabhängigen Standard etabliert.

Bluetooth Profile

Bluetooth bietet des Weiteren verschiedene Profile, die die Ansteuerung durch eine Applikation vereinfachen sollen. Diese Profile übernehmen Teile der Steuerung, wodurch z.B. die Rollenverteilung in der Kommunikation geklärt wird. Zudem benötigt jede Verbindung andere Protokollteile, so dass durch die Einführung von verschiedenen Profilen eine Unterteilung erreicht wird. Der Abbildung 2.5 kann eine Übersicht entnommen werden, auf die nun genauer eingegangen wird.

Als allgemeine Grundlage aller angebotenen Profile dient Generic Access Profile (später auch GAP genannt). Darauf setzen verschiedene Protokolle auf, auf denen wiederum für einzelne, spezielle Funktionen weitere Profile aufbauen. Zum Auffinden von Geräten wird Service Discovery Application Profile¹² eingesetzt. Hierdurch wird SDP gesteuert, so dass die einzelnen Dienste der Verbindungsteilnehmer ausgetauscht werden.

Ein weiteres Profil, welches direkt auf GAP aufbaut, ist das TCS-BIN-based Profil. Wie der Name schon vermuten lässt, können durch dieses Profil Telefonverbindungen gesteuert werden. Als drittes und letztes Profil baut Serial Port Profile¹³ direkt auf GAP auf. Dieses bietet eine Ansteuerungsmöglichkeit für COM-Verbindungen. Da diese Verbindungsart Grundlage für viele höher liegende Protokolle ist, bauen mehrere weitere Profile auf SPP auf.

Es würde an dieser Stelle zu weit führen, alle aufbauenden Profile aufzuzählen und zu beschreiben. Aber die Abbildung 2.5 bietet einen guten, ersten Einblick. Allerdings sind hierbei nur wenige Profile enthalten, die aber insgesamt die wichtigsten sind. Bei tiefergehendem Interesse wird dem Leser entsprechende Literatur nahe gelegt (z.B. Wollert (2002)).

¹²Dieses Profil wird allgemein mit SDAP abgekürzt.

¹³Allgemein wird dieses Profil mit SPP abgekürzt.

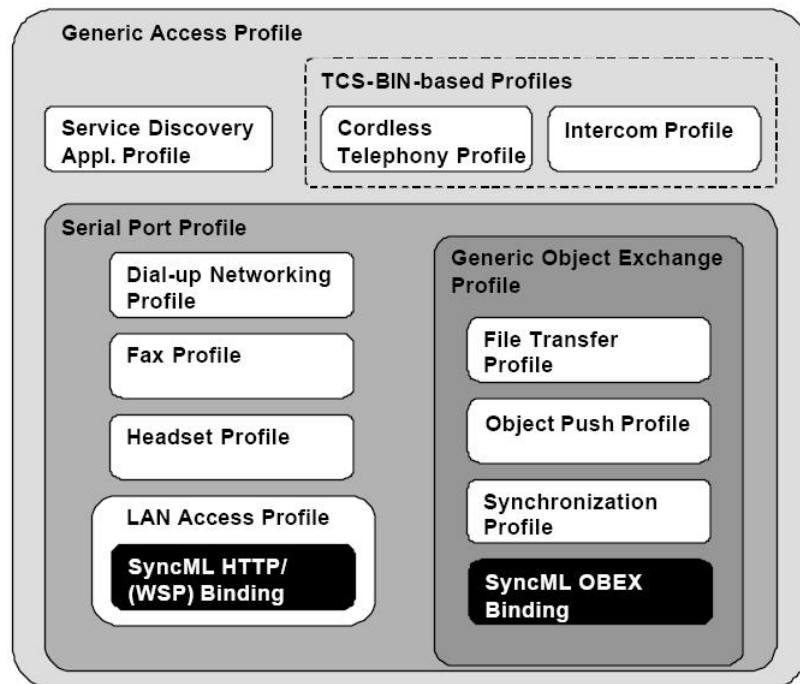


Abbildung 2.5.: Bluetooth-Profile

2.3. WLAN

Mit WLAN steht dem Nutzer eine Funktechnik zur Verfügung, die kabelgebundene Netzwerke ersetzen bzw. lokale flexibler machen sollen. Da diese Technik höhere Datenraten bietet als Bluetooth, wird sie in dieser Arbeit als Verbindung zum Server genutzt. Inzwischen gibt es verschiedene WLAN-Standards sowie unterschiedliche Topologien, die im Folgenden kurz beschrieben werden. In der Literatur sind einige Werke zu finden, durch die man sich einen tieferen Einblick verschaffen kann (siehe Kauffels (2003)). Des Weiteren soll auf die Diplomarbeit von Patrick Postel, Sebastian Schünemann und Jaroslaw Zdrzalek verwiesen werden, die sich ausgiebig mit WLAN-Systemen beschäftigt haben (siehe Postel u. a. (2004)). Da im Laufe der Arbeit diese Technik nur benutzt wird, wird hier ein allgemeiner Überblick gegeben.

Standards

Im Laufe der Zeit sind verschiedene Standards für die WLAN-Technologie entwickelt worden. Diese nutzen zwei unterschiedliche Frequenzbereiche und bieten verschiedene Bandbreiten (siehe auch Schwarz (2004)). Um dem Leser einen kurzen Überblick zu verschaffen, werden

diese Standards kurz verglichen. Allerdings werden weitere Standards entwickelt, wobei hier nur auf aktuell verfügbare Möglichkeiten eingegangen wird.

Als günstigere Variante gibt es die Standards IEEE 802.11b und g. Da hierbei allerdings das weit verbreitete ISM-Band im 2,4GHz-Bereich genutzt wird, treten des Öfteren Konflikte mit anderen Funktechniken wie z.B. Bluetooth auf. Die weite Verbreitung dieses Bereichs ist auf die relativ günstigen Hardwarekomponenten sowie die lizenzfreie Nutzung zurückzuführen. Die ältere der beiden Spezifikationen 802.11b bietet mit 11Mbps die günstigste und zur Zeit weit verbreitetste WLAN-Variante. 802.11g wird inzwischen bei vielen neuen Produkten angeboten, da hiermit bis zu 54Mbps möglich sind.

Im 5GHz-Bereich des ISM-Bands sind IEEE 802.11a sowie 802.11h zu finden. Da allerdings für die Funkübertragung die Hardwarekomponenten zur Zeit noch relativ teuer sind, sind diese Standards noch eher Nischenprodukte. Dagegen konnte auch die lizenzfreie Nutzung bisher nichts ändern. Die Variante a bietet eine Bandbreite von bis zu 11Mbps, was 802.11h mit 54Mbps übertrifft. Für diese Arbeit werden Produkte dieser Spezifikationen nahe gelegt, da in Zusammenarbeit mit Bluetooth Probleme auftreten können. Auf dieses Thema wird später näher eingegangen.

Zudem muss noch darauf hingewiesen werden, dass WLAN-Standards auch dann kompatibel zu einander sind, wenn sie zwar unterschiedliche Geschwindigkeiten unterstützen aber im gleichen Frequenzbereich arbeiten. Dabei wird die Geschwindigkeit dem langsameren Standard angepasst, so dass IEEE 802.11b mit g und 802.11a mit h eingesetzt werden können.

Topologien

Da durch WLAN flexible Verbindungen möglich sind, gibt es auch hier zwei verschiedene Topologien. Zum Einen wird eine feste Verbindung genutzt, wie es schon in drahtgebundenen Netzwerken üblich ist. Dabei stehen die einzelnen Netzwerkteilnehmer in Verbindung zu mindestens einem Access Point, welcher eine zentrale Station bietet. Hier werden Neuzugänge kontrolliert und gesteuert, so dass ein geregelter Ablauf möglich ist. Zudem steuert ein Access Point die Kommunikation auch zu entfernten Teilnehmern. Somit ist eine Verbindung zwischen zwei Teilnehmern möglich, die nicht in direkter Reichweite zueinander stehen.

Als Alternative dazu gibt es Ad-hoc-Netzwerke, bei denen kein Access Point genutzt wird. Zwei einzelne Teilnehmer bauen mit dieser Variante eine direkte Verbindung zueinander auf. Die Kommunikation läuft nach festgelegten Richtlinien ab, die allerdings das Netzwerk unflexibler machen als bei Verbindungen über Access Points. Zudem ist die Reichweite eingeschränkter, da Teilnehmer nur den direkten Umkreis ansprechen können.

Funktionsweise

Anders als bei Bluetooth wird bei WLAN der genutzte Frequenzbereich in Kanäle unterteilt. Alle Teilnehmer eines Netzes müssen den gleichen Kanal nutzen. Zudem wird ein solches Netzwerk durch eine SSID gekennzeichnet, um eine eindeutige Identifizierung zu ermöglichen.

Bei den Standards 802.11b und g stehen dafür 3 und bei den Standards a sowie h 8 Kanäle zur Verfügung. Dadurch können entsprechend viele Netzwerke parallel betrieben werden. Falls mehr Netzwerke eingesetzt werden, als Kanäle verfügbar sind, können entsprechende Störungen auftreten, welche die Datenrate verringern.

2.4. Wechselwirkungen zwischen Bluetooth und WLAN

Da sowohl Bluetooth als auch WLAN im 2,4-GHz-Bereich arbeiten¹⁴, kommt es dadurch zu gegenseitigen Störungen (siehe Zivadinovic (2005)). Da beide Techniken unterschiedliche Strategien zur Bandspreizung nutzen, stören sie sich dabei. Fast Frequency Hopping von Bluetooth wirkt im WLAN wie ein Zerhacker, so dass die Nutzungsrate bis auf 50% absinken kann.

Um dieses Problem zu umgehen, gibt es zwei Möglichkeiten. Zum Einen kann man Bluetoothkomponenten einsetzen, die den Bluetoothstack ab Version 1.2 bieten. Diese beinhalten AFH, welches gestörte Frequenzen erkennt und aus dem Sprungmuster streicht. Somit wird die gegenseitige Störung verringert. Die andere Möglichkeit zielt auf die WLAN-Komponenten ab. Diese sollten IEEE 802.11a oder 802.11h nutzen, da hierbei das 5-GHz-Band genutzt wird. Somit wird Bluetooth ausgewichen. Zwar sind diese Komponenten momentan noch wesentlich teurer, aber gerade bei diesem Szenario mit den evtl. hohen Datenraten, sollte man diesen Bottleneck vermeiden.

Die konsequenteste Variante ist natürlich beide einzusetzen. Dadurch werden sowohl Störungen untereinander als auch durch systemfremde Geräte minimiert, was durch erhöhte Übertragungsraten zum Komfort beitragen und Bottlenecks vermeiden wird.

¹⁴Die Spezifikationen IEEE 802.44b und g arbeiten im 2,4-GHz-Bereich.

2.5. Java

Java ist eine weit verbreitete Programmiersprache, die von der Firma Sun Microsystems¹⁵ frei zur Verfügung gestellt wird. Sun Microsystems unterteilt diese Sprache in verschiedene Anwendungsgebiete, die dafür entsprechende Funktionalität bieten. Zudem ist der Umfang angepasst, so dass eine sinnvolle Unterteilung für die Einsatzgebiete entsteht¹⁶. Da die Programmiersprache allgemein bekannt ist, werden im folgenden Abschnitt die besonderen Einschränkungen von J2ME beschrieben sowie eine weitere Unterteilung namens Personal-Java, welche für den Einsatz auf PDAs gedacht ist.

J2ME

Java 2 Platform, Micro Edition (siehe various (2003a)) bietet ein Bibliothekenpaket, welches speziell auf mobile Systeme ausgelegt ist. Da diese Systeme oft sehr begrenzte Ressourcen bereitstellen, finden sich im Vergleich zu J2SE viele Einschränkungen. Allerdings würde es an dieser Stelle zu weit führen, eine komplette Einführung zu liefern. Daher werden kurz die wichtigsten Merkmale und Unterschiede erläutert. Für tiefergehende Informationen wird auf entsprechende Literatur verwiesen (z.B. Mahmoud (2002)).

Da J2ME überwiegend auf Handys eingesetzt wird, hat Sun Microsystems diese besonders berücksichtigt. Durch die Einführung vom Mobile Information Device Profile (auch MIDP genannt) wird dem Programmierer eine übersichtliche Schnittstelle geliefert, die eine angepasste Oberfläche des Programms zu implementieren zulässt. Somit muss eine entsprechende Klasse¹⁷ durch relativ wenig Sourcecode um die Funktionalität erweitert werden, so dass dem Anwender eine typische Menüführung geboten wird.

Zudem beinhaltet J2ME eine weitere, große Besonderheit, die wegen der begrenzten Ressourcen nötig ist. Da nicht in allen Applikationen jegliche Funktionalitäten genutzt werden, hat Sun Microsystems verschiedene optionale Pakete definiert und spezifiziert, die bei Bedarf eingebunden werden. Dadurch lassen sich auch sehr systemabhängige Funktionen direkt per Java ansteuern, was z.B. Bluetoothverbindungen oder Adressbuchzugriffe möglich macht¹⁸. Die entsprechenden Pakete werden von den jeweiligen Hardwareanbietern bereitgestellt, so dass eine optimale Hardwareansteuerung gegeben ist.

¹⁵<http://www.sun.com/>

¹⁶In Serverapplikationen wird J2EE eingesetzt, Heimanwendungen nutzen J2SE und mobile Software setzt auf J2ME auf

¹⁷In J2ME nennt Sun Microsystems diese MIDlet

¹⁸Eine Übersicht von angebotenen Paketen finden Sie unter <http://java.sun.com/j2me/>

PersonalJava

Durch die Entwicklung von immer leistungsfähigeren PDAs, bietet sich inzwischen die Möglichkeit ein umfangreicheres Javapaket als J2ME auf diesen zu nutzen. Somit werden viele Einschränkungen aufgehoben, so dass die Entwicklung von Software wieder komfortabler wird. Außerdem bieten PDAs eine andere Menüführung, was ein weiterer Grund für die Umstellung ist.

PersonalJava ist eine weitere Spezifikation von Sun Microsystems, welche genau für den eben beschriebenen Fall entwickelt wurde (siehe z.B. various (2003b)). Leider gibt es für dieses Paket relativ wenige Entwicklungstools, was an der momentan geringen Verbreitung liegen könnte. Allerdings bietet PersonalJava den gleichen Funktionsumfang wie Java 1.1.8. Somit kann sich der Entwickler selbst helfen, indem er diese J2SE-Version installiert und nutzt. Zudem lässt sich durch das Programm JavaCheck¹⁹ direkt überprüfen, welche Java-version das entwickelte Programm benötigt. Somit ist eine indirekte Entwicklungsunterstützung gegeben. Mit geringen Anpassungen lässt sich sogar Java Swing nutzen²⁰, wodurch eine komfortable GUI-Entwicklung möglich wird.

Ein weiterer Pluspunkt, der für PersonalJava spricht, ist vergleichbare Nutzung. Da es an sich von der Funktionalität her eine veraltete Version von J2SE ist, braucht sich der Entwickler keiner größeren Umstellung bzw. Einarbeitung zu unterziehen. Zwar sind lange nicht alle Komponenten der aktuellen Version enthalten, aber erfahrene Entwickler finden schnell Alternativen, die schon in dieser Version enthalten sind. Somit geht die Entwicklung schneller voran als mit J2ME.

2.6. Verteilte Systeme

Wenn in einem System verschiedene Aufgaben von unterschiedlichen Systemen erledigt werden, nennt man dieses ein verteiltes System. Es gibt verschiedene Mechanismen zur Nutzung von verteilten Systemen, weswegen hier ein paar kurz beschrieben werden. Sowohl Java RMI als auch CORBA bieten einen solchen Mechanismus.

Eine weitere Entwicklung im Bereich von verteilten Systemen sind Web-Services. Diese bieten Dienste, die eine allgemein definierte Schnittstelle besitzen. Dadurch ist es möglich, die Entwicklung der einzelnen Komponenten komplett zu trennen. Da dieses ein System wesentlich flexibler macht, wird als letztes der Mechanismus SOAP beschrieben.

¹⁹Dieses Programm läuft unter Java und wird direkt von Sun Microsystems angeboten (<http://java.sun.com/products/personaljava/javacheck.html>)

²⁰Eine entsprechende Anleitung findet man z.B. hier http://www.blueboard.com/j2me/notes/2002_7_26.htm

Java RMI

Java bietet eine Möglichkeit, Methodenaufrufe entfernter Objekte zu realisieren. Dieses nennt sich Remote Method Invocation (im Folgenden als RMI bezeichnet) und ist Teil der aktuellen J2SE. Objekte, die für entfernte Aufrufe bereitstehen sollen, werden entsprechend angemeldet. Zudem müssen die Klassen verschiedene Interfaces beinhalten, welche die entfernten Methodenaufrufe kontrollieren und bei Fehler und Problemen entsprechende Exceptions melden. Dieses wird durch einen Dämonprozess auf dem Server unterstützt, der einen Nameservice zum Finden und Nutzen entfernter Objekte realisiert. Mehr zu dieser Technologie finden Sie in der entsprechenden Literatur (z.B. SUN (2005)).

Zwar ist Java RMI sehr einfach einzubinden und bietet somit einen leichten und schnellen Einstieg, aber leider ist es auf Java beschränkt. Daher wurde in dieser Arbeit darauf verzichtet. Außerdem trifft die Bezeichnung Web-Service auf RMI nicht ganz zu, da es sich um einen entfernten Methodenaufruf handelt. Durch die Beschränkung in der Programmiersprache ist ein solcher Dienst nur sehr schwierig in anderen Systeme einzubinden. Das Haupteinsatzgebiet von Java RMI liegt in verteilten Systemen.

CORBA

Durch die Nutzung von CORBA ist es möglich, plattformübergreifende Schnittstellen auf entfernten Ressourcen zu nutzen und auf entfernte Objekte zuzugreifen. Die Object Management Group bietet eine objektorientierte Middleware, welche über einen ORB die Kommunikation zwischen den Netzkomponenten steuert. Dadurch ist eine Nutzung von Objekten in einem Netzwerk möglich, ohne zu wissen, wo sich diese Objekte befinden. Zudem bietet CORBA eine weit gefächerte Unterstützung verschiedenster Programmiersprachen, da durch die IDL alle Klassen und Objekte sowie deren Parameter und Datentypen formal spezifiziert werden. Auf dieser Spezifikation basierend wird das Ganze dann auf die entsprechend genutzte Programmiersprache umgesetzt. Weitere Informationen hierzu sowie allgemein zu CORBA finden Sie in entsprechender Literatur (z.B. OMG (2005) oder Eberhart und Fischer (2001)).

Auf CORBA wurde in dieser Arbeit verzichtet, da durch die entsprechende Nutzung eines ORBs sowie die Umsetzung in die entsprechende Programmiersprache zum Einen ein Mehraufwand entsteht. Zum Anderen ist diese Technologie unflexibler als SOAP, da bei einer Umstellung entsprechend mehr Komponenten angepasst werden müssten. Außerdem liegt das Haupteinsatzgebiet genauso wie bei RMI in verteilten Systemen. In dieser Arbeit soll aber einen Dienst angeboten werden, der für möglichst viele verschiedene Systeme einfach nutzbar ist.

SOAP

SOAP²¹ ist ein Protokoll, welches die flexible Einbindung von Web-Services²² ermöglicht. Dazu wird zur Nachrichtendarstellung XML genutzt. Häufig wird der Transport dieser Nachrichten per HTTP realisiert, wobei SOAP auch in diesem Punkt flexibel ist und z.B. FTP, POP3 usw. zulässt. Durch die Nutzung standardisierter Übertragungsprotokolle sowie die freie Auswahl dieser, bietet SOAP den großen Vorteil, in vielen verschiedenen Umgebungen einsetzbar zu sein.

Durch die Nutzung von WSDL werden die Schnittstellen der Web-Services definiert. Dieses beinhaltet sowohl die Übergabeparameter als auch die Rückgabewerte. Durch den Verzeichnisdienst UDDI wird der entsprechende Server identifiziert, so dass die Kommunikation mit diesem aufgenommen werden kann. Für die Kommunikation werden XML-konforme Nachrichten versendet, welche dann an der entsprechenden Gegenstelle empfangen und ausgewertet werden.

Da SOAP sowohl programmiersprachen- als auch plattformunabhängig ist, bietet es ein sehr flexibles Grundgerüst für die Web-Services in dieser Arbeit. Zudem kommt noch hinzu, dass SOAP zukunftsorientiert ist und immer mehr in verschiedenen Bereichen eingesetzt wird. Für tiefere Informationen wird auf entsprechende Literatur verwiesen (z.B. Eberhart und Fischer (2003) oder auch Frank (2003)).

2.7. Zusammenfassung

Durch das Lesen dieses Kapitels sollte nun jeder Leser die benötigten Grundkenntnisse erlangt haben, um dem Verlauf in den weiteren Kapiteln folgen zu können. Neben den genutzten Funktechnologien und ihre gegenseitigen Beeinflussungen sind auch die Unterschiede zwischen den einzelnen Javaversionen erklärt worden. Zudem wurden verschiedene Mechanismen für verteilte Systeme beschrieben, von denen aber im Laufe der Arbeit nur SOAP zum Einsatz kommt. Die beiden anderen Technologien wurde zur Vergleichsmöglichkeit ebenfalls beschrieben.

²¹SOAP war ursprünglich die Abkürzung für Simple Object Access Protocol und später für Service Oriented Architecture Protocol. Inzwischen ist es als eigenständiger Name bekannt und in den USA gesichert.

²²Ein Web-Service bietet einen Dienst an, welcher über definierte Schnittstellen in andere System integriert werden kann.

3. Anforderungen

Im Laufe dieser Diplomarbeit soll ein System entwickelt werden, welches den Besuch eines Museums komfortabler macht und dem Besucher Informationen zu den Ausstellungsstücken näher bringt. Nach der Beschreibung des Bereichs, in dem wir uns in der Informatik bewegen, folgen die allgemeinen Anforderungen, um zu verdeutlichen, welcher Schwerpunkt für das System bei dieser Arbeit wichtig ist. Danach wird erst auf verschiedene Alternativen eingegangen und dabei erklärt, warum diese nicht gewählt wurden. Abschließend folgt die Erläuterung der Anforderungen an das gewählte System.

3.1. Bereich der Informatik

Die Informatik umfasst viele verschiedene Bereiche, die in sehr unterschiedlichen Umgebungen ihren Einsatz finden. Daher bietet dieser Abschnitt eine Orientierungshilfe, um das Umfeld dieser Arbeit zu erkennen. Zum Einen findet sich in dieser Arbeit ein verteiltes System wieder, da verschiedene Systemkomponenten im Zusammenspiel ein neues Gesamtsystem bieten.

Des Weiteren kann man dieses System zu den ubiquitären Systemen zählen, da dem Nutzer nicht deutlich wird, welche weiteren Komponenten hinter dem Bedienelement arbeiten. Allerdings ist es eher am Rand dieses Bereichs anzusiedeln, da durch die Nutzung eines PDAs dem Anwender sofort deutlich wird, dass er ein Computersystem bedient. Diese Tatsache ist bei vielen Systemen dieses Bereichs dem Nutzer nicht sofort erkennbar.

Zudem befinden wir uns im Bereich der mobilen Anwendungen, was schon durch die Nutzung von verschiedenen Funktechnologien deutlich wird. Hierbei kommen WLAN, Bluetooth und RFID zum Einsatz, was sehr zum Komfort beiträgt. Diese Bereiche kommen auch im Master-Projekt der HAW Hamburg im Wintersemester 2005/2006 zusammen zum Einsatz. Dabei soll ein verteiltes System entwickelt werden, welches den Gast eines Ferienclubs unterstützt (siehe various (2005a)). Dabei werden Informations-, Erkennungs- und Abrechnungssysteme kombiniert, um dem Nutzer einen möglichst hohen Komfort zu bieten.

Als weiteres Projekt aus diesem Bereich der Informatik ist Java System RFID Software der Firma Sun Microsystems (siehe various (2005b)). Hierbei handelt es sich um eine Middleware, die die Einführung von RFIDs in Unternehmen erleichtern soll. Darin ist sowohl die Ansteuerung von RFID-Lesegeräten, ein Eventmanagement sowie ein Informationsserver zur zentralen Datenverwaltung enthalten. Allerdings richtet sich Sun Microsystems mit diesem Produkt an größere Firmen, so dass allein für den Einstieg ein fünfstelliger Eurobetrag verlangt wird.

Ein weiteres Beispielprojekt, das dem hier entwickelten Informationssystem sehr nahe kommt, wird in Berlin angeboten. Die Firma eloqu bietet das mobile Informationssystem mobilTour¹ für Touristen an (siehe eloqu (2005)). Dabei werden Touristikinformationen über einen PDA dem Nutzer bereitgestellt, wobei das System für historische Schauplätze der Berliner Mauer eingesetzt wird. Dabei werden über Bluetoothmarken, die an den Standorten angebracht sind, entsprechende Informationen auf den PDA geladen. Die Informationen werden dabei zentral auf einem Server verwaltet, so dass Übereinstimmung zum System dieser Arbeit erkennbar sind.

3.2. Allgemeine Anforderungen

In der heutigen Zeit sind für viele Menschen Museumsbesuche uninteressant. Da das Internet Informationen zu allen Themengebieten anbietet, ist es weit verbreitet, sich über Interessensgebiete durch dieses Medium zu informieren. Auch wenn das direkte Betrachten von Ausstellungsstücken eine ganz andere Informationsmöglichkeit bietet, nutzen immer weniger Menschen diese. Es ist viel bequemer und schneller, sich durch eine Suchmaschine Informationen zu besorgen.

Um den Besuch eines Museums attraktiver zu gestalten, sollten dem Besucher Informationen flexibel angeboten werden. Zudem wäre es von Vorteil, wenn man sich bei Interesse für ein bestimmtes Objekt genauer über dieses informieren könnte. Daher soll dem Besucher eine Möglichkeit geboten werden, seinen Museumsbesuch individueller zu gestalten. Diese Möglichkeiten machen einen Museumsbesuch interessanter. Der Besucher hat mehr Spaß an einer Ausstellung, was auch dem Museum einen Nutzen bringt, da so mit mehr Besuchern zu rechnen ist.

Momentan bieten Museen für den Besucher zwar allgemeine Informationen zu den einzelnen Ausstellungsstücken. Diese sind aber auf angebrachten Informationstafeln direkt bei den Objekten zu finden. Somit ergeben sich verschiedene Begrenzungen in der Nutzung.

¹Die Firma eloqu bietet noch weitere Systeme für ähnliche Einsatzgebiete an. Daher empfiehlt es sich, bei Interesse die Homepage der Firma genauer durchzusehen.

Zum Einen sind diese Tafeln meist relativ klein und es ist nur eine pro Ausstellungsstück zu finden. Dadurch werden Besucher in der Betrachtung dieser eingeschränkt, da nur wenige Menschen gleichzeitig einer Tafel Informationen entnehmen können. Das empfinden Besucher besonders zu den Hauptzeiten als störend, da dadurch der Museumsbesuch unnötig in die Länge gezogen wird.

Ein weiteres Problem findet sich in den statischen Informationen, die man den Tafeln entnehmen kann. Da diese meist in gedruckter Form angebracht sind, ist es sehr zeitaufwändig, diese bei Bedarf zu aktualisieren. Dafür müssen zuerst neue Tafeln entworfen und gedruckt werden. Diese müssen dann direkt am Ausstellungsstück ausgetauscht werden.

Zuletzt bringt die stationäre Anbringung ein Problem mit sich. Sobald ein Besucher Informationen zu einem Ausstellungsstück ansehen möchte, muss er auch zu diesem Objekt gehen. Leider ist das auch der Fall, wenn er nur eine bestimmte Information, wie z.B. der Name des Künstlers, wissen möchte. Da in diesem Fall das Objekt selbst relativ unwichtig ist, empfindet der Besucher den Weg zum diesem als nervig, was sich auch negativ auf den gesamten Museumsbesuch auswirken kann.

3.3. Lösungsansätze

Um diese Probleme zu lösen, ergeben sich verschiedene Lösungsansätze. Einige davon werden hier beschrieben und erläutert, inwieweit dadurch die genannten Probleme gelöst oder umgangen werden können.

Anpassung der Informationstafeln

Das zuerst genannte Problem lässt sich relativ einfach beheben. Wenn die Informationstafeln größer erstellt werden, können mehr Menschen diese gleichzeitig nutzen. Zudem wäre es möglich, bei Bedarf mehrere Tafeln pro Ausstellungsstück anzubringen. Besonders bei großen Objekten wäre das eine Erleichterung für den Besucher Informationen zu bekommen.

Der Vorteil dieser Lösung liegt in der kostengünstigen Realisierung. Zum Einen entstehen dadurch überwiegend nur einmalige Kosten, da es sich um eine einmalige Anbringung handelt. Zudem ist dieses technisch einfach zu realisieren. Zum Anderen kann diese Verbesserung über einen längeren Zeitraum verteilt werden. Dadurch hat das Museum keinen hohen Investitionsbetrag sondern kann, sobald der Finanzplan dieses ermöglicht, einzelne Objekt aufrüsten.

Allerdings werden durch diesen Ansatz die beiden anderen Probleme nicht gelöst. Es werden dem Besucher weiterhin statische Informationen bereitgestellt. Und um diese abzurufen, muss derjenige sich zum Ausstellungsobjekt begeben.

Dynamische Informationstafeln

Eine Alternative zum ersten Ansatz sind dynamische Informationstafeln. Dabei können neben den bisherigen Informationstafeln auch Displays an den Objekten angebracht werden. Diesen Displays kann dann jeder Besucher verschiedene Informationen entnehmen, wobei diese durch ein entsprechendes Eingabegeräte, wie z.B. eine Maus oder ein Touch-Screen, gesteuert werden.

Somit könnte jeder Besucher genau die Information abrufen, die ihn interessieren. Das stellt natürlich einen enormen Vorteil zum ersten Ansatz dar. Denn dieser Service macht einen Museumsbesuch wesentlich interessanter und individueller. Des Weiteren können diese Displays vernetzt werden, so dass die Informationen zentral aktualisiert und gewartet werden können.

Allerdings hat dieser Lösungsansatz verschiedene Nachteile. Zum Einen entstehen durch die benötigten Displays enorme Kosten. Auch der Aufbau des Systems bringt einen weiteren Kostenfaktor, wobei dafür womöglich das Museum dafür geschlossen werden muss. Zudem müssen alle Displays vernetzt werden, damit eine zentrale Wartungsmöglichkeit gegeben ist. Dafür wird entsprechendes Fachpersonal benötigt, so dass durch diesen hohen Kostenaufwand das System für viele Museen uninteressant wird.

Zudem werden nicht alle genannten Probleme gelöst. Wenn dieser Ansatz mit dem ersten kombiniert wird, können zwar viele Menschen Informationen bekommen und bei Interesse diese individuell abrufen. Aber es besteht dann noch immer das Problem, dass die Informationen nur stationär abgefragt werden können.

Mobile Informationstafeln

Ein weiterer Lösungsansatz nutzt mobile Informationstafeln, damit alle genannten Probleme gelöst werden können. Dabei erhält jeder Besucher einen mobile Client. Auf diesem werden verschiedene Informationen zu den einzelnen Ausstellungsstücken dargestellt. Des Weiteren kann der Besucher individuell Informationen abrufen und somit sich nach den eigenen Interessen entsprechend informieren. Zudem kann jeder Client die einzelnen Objekte identifizieren, damit der Besucher unabhängig von anderen Informationsmedien ist.

Des Weiteren können alle Informationen auf einem zentralen Server verwaltet und gewartet werden. Da dieser per Funk abgefragt wird, können Besucher Informationen unabhängig vom Standort abrufen. Dabei ist allerdings zu beachten, dass der Server von jedem Punkt im Museum erreichbar sein sollte, wie es in Abbildung 3.1 exemplarisch dargestellt wird.

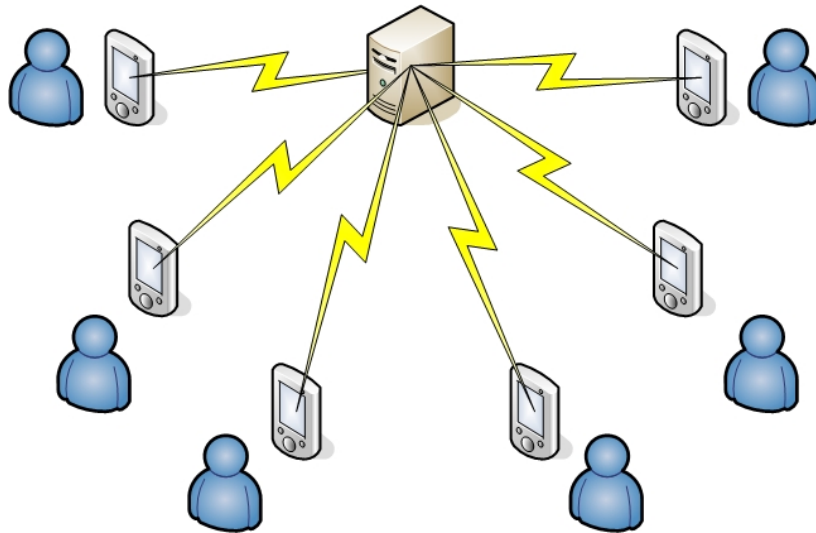


Abbildung 3.1.: Systemaufbau

Außerdem können durch mobile Informationstafeln viele weitere Möglichkeiten geboten werden. Eine Sprachauswahl würde auch sprachfremden Menschen ein Besuch ermöglichen. Des Weiteren kann das System eine Verbindung zum Internet bereitstellen, so dass auch Informationen über eine Suchmaschine abgerufen und bereitgestellt werden können. Und natürlich ist mit einem solchen System möglich, den Besuch durch Multimedia zu unterstützen. Neben verschiedenen Sprachausgaben über einen Kopfhörer sind auch Filme und Animationen denkbar, um dem Besucher Informationen näher zu bringen.

Allerdings bietet dieses System auch Nachteile. Zum Einen sind auch bei dieser Variante hohe Kosten zu erwarten, welche aber über einen längeren Zeitraum verteilt werden können. Zu Beginn wird sowohl ein Server als auch mehrere Clients benötigt. Wenn das System aber als Erweiterung zum aktuellen Stand eingesetzt wird, kann mit der Zeit die Clientanzahl aufgestockt werden. Zwar steht dann dieser Service nicht jedem Besucher zur Verfügung, aber diese Abstufungen macht die Finanzierung für viele Museen leichter.

Des Weiteren entstehen Kosten durch Fachpersonal bzw. durch entsprechende Schulungen. Da es sich um ein flexibles System handelt, sind Fachkenntnisse für die Wartung erforderlich. Allerdings sind diese im laufenden Betrieb überschaubar, da dabei überwiegend Wartungsarbeiten anstehen. Abschließend ist noch darauf hinzuweisen, dass ein entspre-

chender Diebstahlschutz benötigt wird. Gerade bei mobilen Clients ist ein Diebstahl relativ einfach, weswegen ein Schutz, wie z.B. in einem Kaufhaus, unverzichtbar ist.

Zwar mögen jetzt die Kosten hoch erscheinen, aber man muss diese im Zusammenhang mit dem Nutzen sehen. Ein Besuch wird durch einen solchen Dienst wesentlich interessanter. Hinzu kommt die Tatsache, dass bisher kein Museum einen solchen Service bietet. Das bringt eigene Werbung mit sich, so dass auch mit Besuchern zu rechnen ist, welche eher an der Technik als an den Ausstellungen interessiert sind. Zudem sind die Kosten insgesamt über einen längeren Zeitraum verteilbar, was das System finanzierbar machen sollte.

3.4. Anforderungen an das Lösungssystem

Da nur der letzte Lösungsansatz alle beschriebenen Probleme lösen kann, viel die Entscheidung auf diesen. Um aber eine Realisierung zu ermöglichen, muss das System analysiert werden, damit die benötigten Anforderungen erkennbar werden. Dabei ist zu bedenken, dass die weit gefächerten Darstellungsmöglichkeiten den Rahmen dieser Arbeit sprengen würden. Es liegt daher das Hauptaugenmerk auf der Entwicklung eines Prototypen, der eine reine Textdarstellung bietet. Die Analyse dazu ist im nachfolgenden Abschnitt zu finden.

Gesamtsystem

Zuerst müssen die Aktionen des Gesamtsystems, die dem Besucher ermöglicht werden sollen, beschrieben werden. Damit jeder Besucher allgemeine Informationen zu einem Ausstellungsstück erhalten kann, muss das System diese bereitstellen. Dabei ist zu beachten, dass das System dadurch möglichst wenig in der Mobilität eingeschränkt wird. Des Weiteren soll der Besucher tieferegehende Informationen abrufen können, falls Interesse an diesen bestehen. Diese beiden Aktionsmöglichkeiten sind in den Abbildungen 3.2 und 3.3 abzulesen.

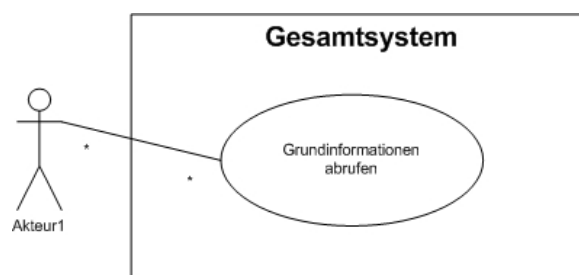


Abbildung 3.2.: Use Case: Grund Informationen

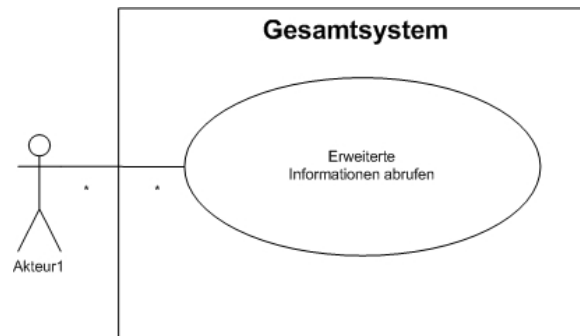


Abbildung 3.3.: Use Case: Erweiterte Informationen

Um dem Nutzer Informationen bereitstellen zu können, benötigt dieser einen mobilen Client für die Darstellung. Des Weiteren soll mit jedem Client eine eindeutige Identifizierung des Objektes möglich sein. Zudem ist eine zentrale Verwaltung und Wartung der Informationsdaten wichtig. Hierfür wird ein Server benötigt, der über ein Netzwerk den einzelnen Clients entsprechende Informationen liefert.

mobiler Client

Der mobile Client wird während des Besuchs vom Nutzer mitgeführt. Ein Notebook o.Ä. wäre ungeeignet, da dieses zu schwer und umständlich wäre, um es durchgehend mitzuführen. Besser geeignet ist daher ein PDA, auch wenn dieser ein recht kleines Display bietet. Zudem sollte dieser PDA Funkverbindungen bieten, um eine Kommunikation mit dem Server zu ermöglichen.

Des Weiteren muss jeder Client die einzelnen Ausstellungsobjekte identifizieren können. Dieses sollte drahtlos geschehen, damit der Besucher nicht in seiner Mobilität eingeschränkt und die Nutzung des Gesamtsystems komfortabler wird. Abschließend ist noch zu sagen, dass ein entsprechender Museumsbesuch mehrere Stunden dauern kann, weswegen eine gute Akkuleistung wichtig ist.

Objektidentifikation

Zur Identifikation einzelner Ausstellungsobjekte sind verschiedene Technologien verfügbar. Um einen möglichst hohen Komfort zu bieten, sollen Funkidentifikationen eingesetzt werden. Zwar wären auch Barcodes mit entsprechenden Lesegeräte denkbar, aber für diese Technik wird ein Sichtkontakt zum Strichcode benötigt. Dieses ist allerdings problematisch, wenn

mehrere Besucher zum gleichen Objekt Informationen abrufen wollen. Aus dem gleichen Grund sind Techniken mit Infrarot nicht einsetzbar.

Des Weiteren sollten die eingesetzten IDs keine Stromquelle benötigen. Dadurch lässt sich diese Fehlerquelle beseitigen und ein dauerhafter Einsatz ist möglich. Zudem wird eine mögliche Wartung minimiert, da auf Akkuleistungen bei den Identifikationsmarkierungen nicht geachtet werden muss. Durch diese Einschränkung fallen allerdings Techniken wie z.B. Bluetooth nicht mehr ins Blickfeld für die benötigte Lösung.

Die genannten Einschränkungen bieten die Nutzung von RFIDs nahezu an. Dabei können passive Tags eingesetzt werden, welche an den einzelnen Objekten angebracht werden. Durch ein entsprechendes Lesegerät können dann IDs ausgelesen und über den PDA weiterverarbeitet bzw. weitergeleitet werden.

Server

Um den Clients die entsprechenden Informationen bereitzustellen, wird ein Server benötigt, welcher eine Datenbank bereitstellt. Wichtig ist auch hier bei, dass dieser Server eine Funkverbindung zu den Clients aufbauen kann, umso die Mobilität der Clients zu garantieren. Da die Datenbank allerdings wahrscheinlich nicht im Dauerbetrieb genutzt werden muss², gibt es wenig Einschränkungen an die Datenbank. Das Erhöht die Auswahl in diesem Bereich enorm.

3.5. Zusammenfassung

In diesem Kapitel wurden die allgemeinen Anforderungen an das zu entwickelnde System beschrieben. Auf diese Beschreibung folgen einige mögliche Lösungswege, wobei deren Vor- und Nachteile aufgezeigt werden. Nach der Begründung für die ausgewählte Lösung folgen speziellere Anforderungen, die an diese Lösung gestellt werden. Dabei wird sowohl auf das Gesamtsystem als auch auf die einzelnen Komponenten eingegangen.

²Schließlich hat ein Museum feste Öffnungszeiten, so dass für Updates und Wartungen das System auch heruntergefahren werden kann.

4. Design

In diesem Kapitel wird sich mit dem Design des Informationssystems auseinandergesetzt. Dazu werden sowohl das Design des Gesamtsystems als auch das der Komponenten erläutert. Nachdem das Gesamtsystem beschrieben wird, folgt eine Designbeschreibung der Kommunikation zwischen den Systemkomponenten. Abschließend wird auf das Design der Einzelkomponente genauer eingegangen.

Im Vordergrund des Designs steht die Flexibilität des Ganzen. Nur so lässt sich das System an entsprechende Umgebungseinschränkungen und -veränderungen anpassen, ohne dafür einen enormen Mehraufwand bewältigen zu müssen. Dadurch lässt sich das Programm in verschiedensten Umgebungen einsetzen, was ein klarer Vorteil zu festgelegten und eingeschränkten Designs darstellt. Hinzu kommt ein entsprechender Komfort, der dem Nutzer geboten werden soll. Dadurch sollen viele Aktionen auch zwischen den Komponenten überwiegend automatisch ablaufen, so dass gerade auch Nutzer, die keinen tieferen Informatikwissensstand haben bzw. denen nur das Vertrauen in die Technik fehlt, das System sinnvoll einsetzen können.

4.1. Gesamtsystem

Im Folgenden wird das Gesamtsystem auch als MuseumAssistant bezeichnet. MuseumAssistant bieten dem Nutzer zwei Aktionsmöglichkeiten (siehe Abbildung 4.1). Um sich einen ersten Eindruck über das Ausstellungsobjekt zu machen, kann der Nutzer Grundinformationen abrufen. Dabei werden einige kurze Angaben dazu angezeigt. Falls diese nicht ausreichend sein sollten, können noch erweiterte Informationen abgerufen werden. Hierbei handelt es sich meist um einen längeren Text, der verschiedene Aspekte des Objekts erläutert.

Um eine möglichst große Mobilität zu erreichen, wurde MuseumAssistant in zwei Hauptkomponenten unterteilt, wie es auch schon im vorherigen Kapitel erläutert wurde. Dabei übernimmt ein PDA die Aufgaben des Clients. Ein Server stellt eine zentrale Zugriffsmöglichkeit auf eine Datenbank bereit, welche die Informationsdaten beinhaltet. Zudem nutzt der Client ein RFID-Lesegerät zur Identifikation der Ausstellungsstücke. Diese wird durch eine Funkverbindung angesteuert, um entsprechende Kabel zu vermeiden.

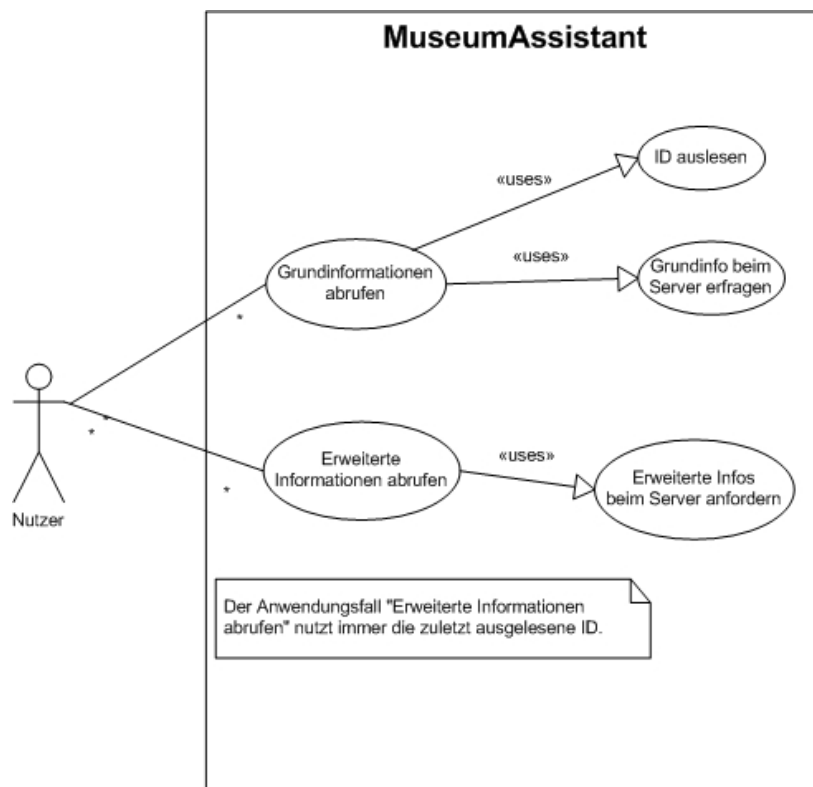


Abbildung 4.1.: Use Case Diagramm

4.2. Clientsoftware

Auf dem Client wird eine Software benötigt, die sowohl als Schnittstelle zum Nutzer als auch der Kommunikation zu den anderen Komponenten des Systems dienen soll. Dafür wird das System entsprechend unterteilt, wie es auch der Abbildung 4.2 zu entnehmen ist.

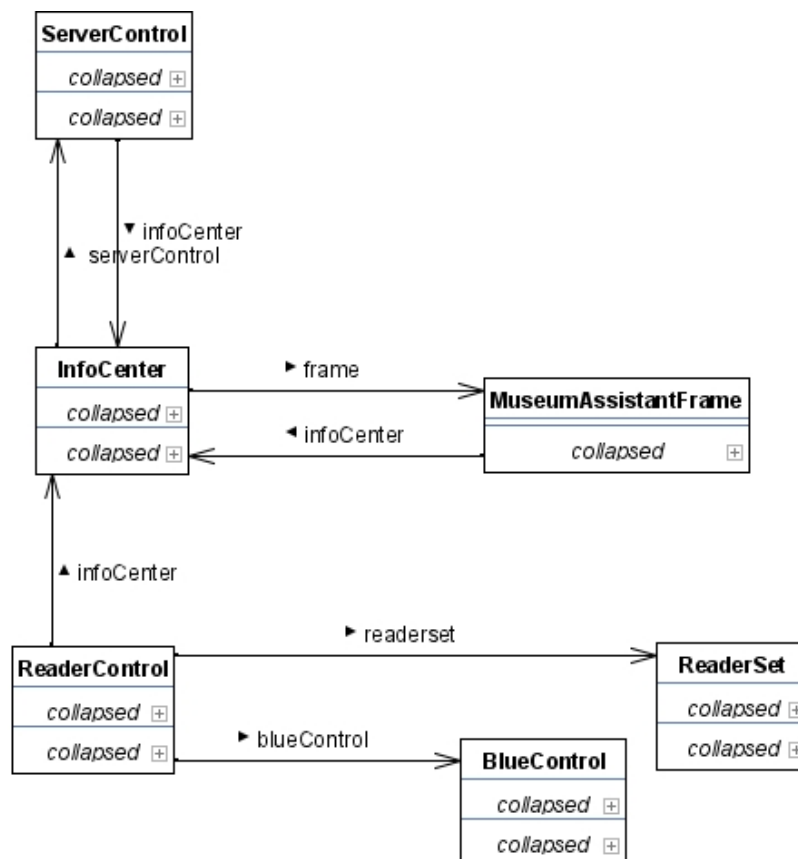


Abbildung 4.2.: Klassendiagramm: Clientsoftware

Hierbei nimmt die Klasse InfoCenter eine zentrale Steuereinheit ein. Wie schon der Name vermuten lässt, bietet die Klasse MuseumAssistantFrame die GUI als Schnittstelle zum Nutzer. Die Klassen ServerControl sowie ReaderControl übernehmen jeweils die Ansteuerung der entsprechenden Systemkomponente. Hierbei nutzt ReaderControl für die Kommunikation per Bluetooth die Klasse BlueControl. Des Weiteren sind die Steuerungskommandos des RFID-Lesers in der Klasse ReaderSet zu finden.

Im Folgenden wird zuerst auf die Art der Informationsdarstellung eingegangen. Dabei wird kurz verdeutlicht, wo der Schwerpunkt in diesem Entwurf liegt. Danach wird der Mechanis-

mus, der zur Identifikation des Lesegeräts dient und die Ansteuerung desselben ermöglicht, genauer betrachtet.

Informationsdarstellung

Bei der Betrachtung des Gesamtsystems in der im Szenario beschriebenen Umgebung kommt der Darstellung der Informationen eine besondere Rolle zu. Hier besteht die aktive und bewusste Nutzung des System durch den Anwender, der nicht weiß, welche Aktionen im Hintergrund ablaufen. Daher muss diese Schnittstelle einfach in der Bedienung sowie übersichtlich sein.

Aus diesem Grund wird in diesem Design die gesamte GUI auf zwei mögliche Aktionen, die durch entsprechende Buttons ausgelöst werden, und einem Darstellungsfeld beschränkt. Zudem ist noch zu sagen, dass der Schwerpunkt der Arbeit auf den im System ablaufenden Hintergrundaktionen und die dadurch erreichte Nutzbarkeit des Gesamten liegt. Daher wird vorerst eine reine Textdarstellung genutzt, die aber bei entsprechenden Umgebungsanforderungen angepasst werden kann.

Lesegerätidentifikation

Ein besonderer Schwerpunkt dieser Arbeit liegt in den automatisierten Abläufen. Um diese möglichst flexibel zu halten, beinhaltet das System einen Identifikationsmechanismus, durch welchen das System selbst erkennen kann, welche Hardware als Lesegerät genutzt wird. Hierfür werden für alle in Frage kommenden Geräte XML-Dateien erstellt, welche komplette Kommandosätze beinhalten.

Ein Kommando ist eine Zeichenabfolge, welche von einem Kommunikationsteilnehmer an den anderen gesendet wird. Da aber bei einer Verbindung meist eine Folge von Kommandos zwischen den Teilnehmern ausgetauscht werden, werden diese als eine Kommandosequenz bezeichnet. Eine solche Sequenz besteht aus einer Anzahl von Kommandos in einer festgelegten Reihenfolge, wobei beide Seiten senden und empfangen können. Zudem erreicht man mit einer Kommandosequenz eine bestimmte Aktion eines Teilnehmers wie z.B. das Auslesen einer ID. Mehrere Kommandosequenzen werden dann als ein Kommandosatz zusammengefasst. In einem Satz von Kommandos finden wir mindestens eine Identifikationssequenz und eine Auslesesequenz. Zudem können weitere Sequenzen in einem Satz hinterlegt werden, worauf später genauer eingegangen wird.

Alle Kommandosätze werden in einem entsprechenden Verzeichnis abgelegt, welches der Mechanismus abarbeitet. Ein solcher Kommandosatz beinhaltet eine Kommandosequenzen, welche der eindeutigen Identifizierung der Hardware dient, sowie die Kommandosequenz

zum Auslesen einer ID. Sobald die Identifikationssequenz fehlerfrei abgearbeitet wird, beendet das System den Identifikationsmechanismus und stellt der Clientsoftware die Ansteuerungskommandos bereit. Der genaue Ablauf wird im Zustandsdiagramm 4.3 beschrieben.

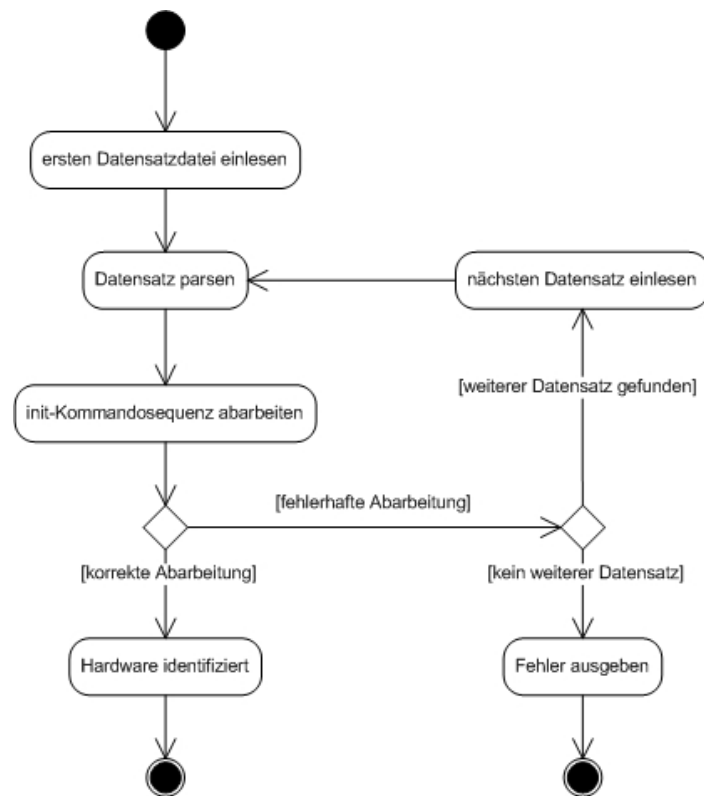


Abbildung 4.3.: Zustandsdiagramm: Identifikationsdiagramm

Der Vorteil in der Nutzung dieses Mechanismus liegt darin, schnell und einfach die Hardware zur Objektidentifikation austauschen zu können. Zudem wurde die DTD mit möglichst wenig Einschränkungen realisiert, damit sehr viele verschiedene Arten von Protokollabläufen dargestellt werden können. Somit ist auch der Verzicht auf RFIDs in diesem Zusammenhang möglich, falls eine andere Technologie besser in die bestehenden Umgebungsvorgaben passt.

4.3. Serversoftware

Der Server wird als zentrale Datenverwaltung benötigt. Hier sollen die Informationsdaten gewartet und den Clients bereitgestellt werden. Allerdings muss dieses nicht zwingend durch

eigene Hardware realisiert werden, wenn schon Rechner andere Serveraufgaben übernehmen. Daher liegt es nahe, eine verbreitete Technik für den Dienst zu nutzen und hierfür einen entsprechenden zu entwickeln. Deshalb wird ein Web-Service für diesen Dienst genutzt.

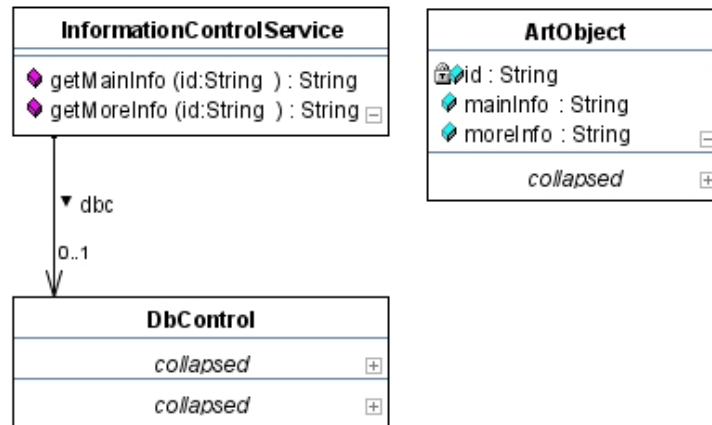


Abbildung 4.4.: Klassendiagramm: Serversoftware

Der Abbildung 4.4 ist die Struktur der Serversoftware zu entnehmen. Die Klasse ArtObject stellt dabei ein Ausstellungsstück dar und beinhaltet alle notwendigen Informationen. Des Weiteren ist die Klasse InformationControlService enthalten, welche die Schnittstelle des Dienstes darstellt. Die hierfür benötigten Methoden werden durch den Server im Netzwerk bereitgestellt. Abschließend ist noch zu sagen, dass die Klasse DbControl als Schnittstelle für den Zugriff auf die Datenbank dient. Beim Start dieses Dienstes wird auch hier die Initialisierung der Datenbank realisiert, damit ein Zugriff im laufenden Betrieb möglich ist.

Darauf folgt eine kurze Erläuterung der Vorteile eines Web-Services sowie einer Datenbank im Zusammenhang mit diesem System. Zwar sind diese Aussagen relativ allgemein gehalten, aber im Kapitel der Realisierung wird näher auf dieses Thema eingegangen.

Web-Service

Einen Web-Service kann man mit verschiedenen Verbindungsarten abrufen und er wird als Erweiterung eines Webserver eingesetzt. Zudem ist es eine zukunftsweisende Technik, die immer größere Verbreitung erreicht und auch durch andere Clients genutzt werden kann. Somit kann ein solcher Dienst in andere Systeme zusätzlich integriert werden, ohne dafür Änderungen an meinem System vornehmen zu müssen. Dafür muss nur der Webserver, der den Dienst anbietet, erreichbar sein.

Die anderen genannten Möglichkeiten für verteilte Systeme sind hier eher unpassend. Bei RMI ist man an die Programmiersprache Java gebunden. Dadurch wird die Flexibilität von MuseumAssistant sehr eingeschränkt. Auch der Server wird sehr eingeschränkt, da es sich nicht um einen allgemein abrufbaren Dienst sondern um eine Gegenstelle in einem Projekt handelt. Ein ähnliches Problem entsteht bei der Nutzung von CORBA. Hierfür wird eine weitere Systemkomponente, den ORB, benötigt. Da diese Einschränkungen durch einen Web-Service mit SOAP nicht entstehen, fällt die Entscheidung auf diese Technik.

Durch festgelegte Definitionen kann dieser Dienst abgerufen werden. Da dieses durch eine Message geschieht, kann eine solche Abfrage auch aus anderen Programmiersprachen erfolgen. Dabei muss nur die entsprechende Nachricht erstellt und abgeschickt werden. Der Server antwortet dann mit einer entsprechenden Nachricht, der die gewünschten Informationen entnommen werden können. Im Allgemeinen wird ein Web-Service durch UDDI bereitgestellt. Dieser Verzeichnisdienst wird aber für MuseumAssistant nicht eingesetzt, da der hier genutzte Dienst nur lokal Anwendung findet.

Datenbank

Da eine Vielzahl von Datensätzen für das System benötigt wird und diese zentral gewartet werden sollen, liegt es Nahe, diese in einer Datenbank zu verwalten. Ein Datensatz besteht aus einer Grundinformation sowie einem Text, welcher weitere Informationen liefern soll. Daher ist vorerst eine einfache Variante einer Datenbank ausreichend. Zudem handelt es sich überwiegend um Abfragen, so dass bei einem Performancevergleich darauf besondere Rücksicht genommen werden sollte.

Was die Auswahl außerdem sehr frei hält ist die Tatsache, dass die Wartung nicht im laufenden Betrieb stattfinden muss. Da MuseumAssistant nur während der Öffnungszeiten des Museums laufen muss, können entsprechende Wartungsarbeiten außerhalb dieser Zeiten erledigt werden. Diese Gründe machen klar, dass nur wenige Datenbanken diese Vorgaben nicht erfüllen. Deshalb wird in dieser Arbeit eine einfache Datenbank genutzt, zu der später mehr gesagt wird. In einem bestehenden System lässt sich MuseumAssistant auch entsprechend anpassen, damit evtl. schon vorhandene Datenbanksysteme genutzt werden können.

4.4. Kommunikation

Es gibt im System zwei Arten der Kommunikation. Hierbei unterscheiden wir zwischen der Ansteuerung des RFID-Lesegeräts vom PDA und der Informationsabfrage des PDAs beim

Server. Erstes geschieht nach einem Protokoll, welches vom Hersteller des Lesegeräts definiert wurde. Im Gegensatz dazu wurde die Serverabfrage über einen Web-Service definiert, was die Kommunikation zwischen dem PDA und dem Server flexibler macht. Dabei werden bei einer Abfrage der Grundinformationen (siehe Abb. 4.5) beide Kommunikationsarten und bei der Abfrage der erweiterten Informationen (siehe Abb. 4.6) nur die Client-Server-Kommunikation angewendet. Mehr zu den beiden Kommunikationen kommt in den nächsten beiden Abschnitten.



Abbildung 4.5.: Sequenzdiagramm: Abfrage Grundinformationen

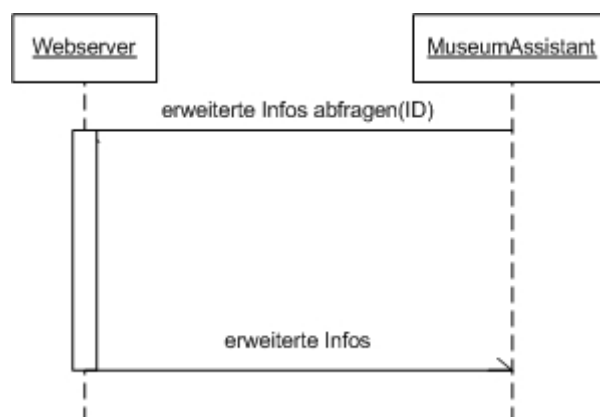


Abbildung 4.6.: Sequenzdiagramm: Abfrage erweiterte Informationen

PDA - RFID-Lesegerät

Die Kommunikation zwischen dem PDA und dem RFID-Lesegerät erfolgt überwiegend durch Kommandos, welche vom Hersteller des Lesegeräts vorgegeben werden. Diese Kommandos werden per Funk ausgetauscht, was mit Bluetooth realisiert werden soll. Bluetooth bietet RFCOMM, welches einen Kabelersatz für die häufig genutzte COM-Schnittstelle bietet. Zudem reicht eine solche Verbindung aus, da bei einer Kommandosteuerung eher niedrige Bandbreiten benötigt werden.

Des Weiteren werden Kommandosequenzen in Dateien abgelegt, welche dann dem System beim Start verfügbar gemacht werden. Dadurch ist es möglich, Anpassungen an der Kommunikation zu vollziehen, ohne dafür den Sourcecode verändern zu müssen. Diese Kommandosequenzen werden nach einer festgelegten Definition dort hinterlegt. Im laufenden Betrieb werden diese geparkt, wodurch sie dann verfügbar sind.

Client - Server

Die Kommunikation zwischen dem Client und dem Server benötigt allerdings höhere Datenraten. Zwar werden in der ersten Version nur reine Textdaten versendet, aber weitere Informationsmedien wie Bilder oder gar Filme sind durchaus umsetzbar. Zudem bietet der Server die Datenbankabfrage per Web-Service an. Durch diese beiden Aspekte wird deutlich, dass eine Netzwerkverbindung (wie z.B. TCP/IP) einen sinnvollen Einsatz ermöglicht.

Um die Mobilität des Clients beizubehalten, empfiehlt sich die Nutzung von WLAN. Damit ist eine kabellose Serverabfrage mit entsprechender Bandbreite möglich. Zudem bietet WLAN eine Netzwerkumgebung wie sie auch schon aus kabelgebundenen Netzen bekannt ist. Dadurch lassen sich viele Techniken direkt auf dieser Netzwerktechnik anwenden, wodurch eine Nutzung sehr komfortabel wird.

4.5. Zusammenfassung

In diesem Kapitel wird sich mit dem Design des Systems befasst. Zuerst wird das des Gesamtsystem beschrieben. Darauf folgen genauere Beschreibungen zu den einzelnen Komponenten, wodurch eine übersichtliche Unterteilung geboten wird. Abschließend wird auf die Kommunikation eingegangen, wobei auch hier eine Unterteilung erfolgt. Dadurch werden die unterschiedlichen Kommunikationsarten einzeln beschrieben.

5. Realisierung

Im Laufe dieser Diplomarbeit wurde das im vorherigen Kapitel beschriebene Design umgesetzt. Die Realisierung wird in diesem Kapitel beschrieben und dabei auf einzelne Aspekte genauer eingegangen.

Im ersten Abschnitt wird sich mit den Hardwarekomponenten befasst. Hierbei erfolgt eine Beschreibung der Hardware, die für den Client benutzt wird und aus welchen Gründen gerade diese gewählt wird. Die Hardware des Servers ist nebensächlich, weswegen auf eine Beschreibung dieser verzichtet wird.

Danach folgt ein kleiner Abschnitt zur genutzten Programmiersprache, wobei auf die Vor- und Nachteile dieser eingegangen und dadurch die Auswahl begründet wird. Auf einen Vergleich mit anderen Programmiersprachen wurde verzichtet, da dieses den Rahmen der Diplomarbeit sprengen würde.

Nach diesem Abschnitt folgt in Kapitel 5.3 die Beschreibung der Clientsoftware. Hierbei wird neben den allgemeinen Erklärungen auch wieder auf die beiden Teilaspekte der Informationsdarstellung sowie der Lesegerätidentifikation eingegangen. Zudem wird sich den Kommandosätzen zugewendet.

Danach folgt eine Beschreibung der Client- und Serversoftware. Hierbei wird in Kapitel 5.4 genauer auf den Teil des Web-Services sowie der Datenbank eingegangen, wie es schon aus dem Designkapitel bekannt ist. Des Weiteren werden in Kapitel 5.3 und 5.4 die genutzten Bibliotheken beschrieben, die für die Realisierung des Ganzen genutzt wurden.

Im abschließenden Unterkapitel wird die Kommunikation zwischen den Komponenten genauer betrachtet.

5.1. Hardwarekomponente

In den nächsten Abschnitten werden die genutzten Hardwarekomponenten beschrieben. Hierbei ist allgemein zu beachten, dass die meisten Komponenten von der HAW Hamburg

bereitgestellt wurden. Dabei wurde auch sehr darauf geachtet, dass schon vorhandene Hardware eingesetzt wird, weswegen nicht immer die idealen Komponenten genutzt werden. Genaueres erfahren Sie dann an entsprechender Stelle in diesem Kapitel.

PDA

Als PDA wird eine iPAQ H5550 von Hewlett-Packard (siehe Abb. 5.1) eingesetzt. Dieser bietet neben dem Betriebssystem Pocket PC 2003 von Microsoft Corporation eine WLAN- (802.11b) sowie eine Bluetoothschnittstelle (Version 1.1). Somit sind die technischen Voraussetzungen für die drahtlose Kommunikation durch die mobile Hardware gegeben.

Der iPAQ wurde von der Hochschule für angewandte Wissenschaften Hamburg zur Verfügung gestellt. Für eine marktreife Nutzung von MuseumAssistant sollte es aber von Vorteil sein, einen günstigeren PDA auszuwählen. Da der iPAQ H5550 mehr Funktionen bietet als für MuseumAssistant benötigt werden, könnten durch eine angepasste Hardwareauswahl Kosten gesenkt werden.



Abbildung 5.1.: iPAQ H5550

Als KVM¹ ist JeodeRuntime der Firma Insignia Solutions installiert, welche für den Ablauf

¹KVM ist eine abgespeckte Java Virtual Maschine, welche für mobile System ausgelegt ist.

der Clientsoftware benötigt wird. Zwar gibt es auch KVMs anderer Anbieter, aber da die vorinstallierte kostenlos ist und den vollen Funktionsumfang bereitstellt, bietet sich die Nutzung an. Zudem bietet dieser PDA weitere Features, wie z.B. ein biometrisches Fingerabdruck-Lesegerät oder eine Speichererweiterung durch SD-Cards. Diese Funktionen werden aber für diese Diplomarbeit nicht benötigt.

Identifikation

Für den Einsatz als vermarktungsfähiges Produkt ist es sinnvoll, ein RFID-Lesegerät zu entwickeln, welches eine Bluetoothschnittstelle bietet. Da diese Entwicklung aber einen zu großen Aufwand für diese Diplomarbeit mit sich bringen würde und nicht im Schwerpunkt liegt, wird als Testobjekt ein Lesegerät der Firma SONMicro (siehe Abb. 5.2) benutzt. Dieses wird über die mitgelieferte COM-Schnittstelle um ein Bluetoothmodul der Firma PicoPlug (siehe Abb. 5.3) erweitert, damit eine entsprechende Verbindung möglich ist.

Die Kommunikation zwischen Lesegerät und RFID-Tag läuft bei diesem Modul im Frequenzbereich von 125kHz. Dabei ist eine Reichweite von ca. 12 cm verfügbar, welche für die Testumgebung vorerst ausreichend ist. Für den späteren Einsatz wird allerdings eine höhere Reichweite benötigt, worauf an anderer Stelle genauer eingegangen wird.

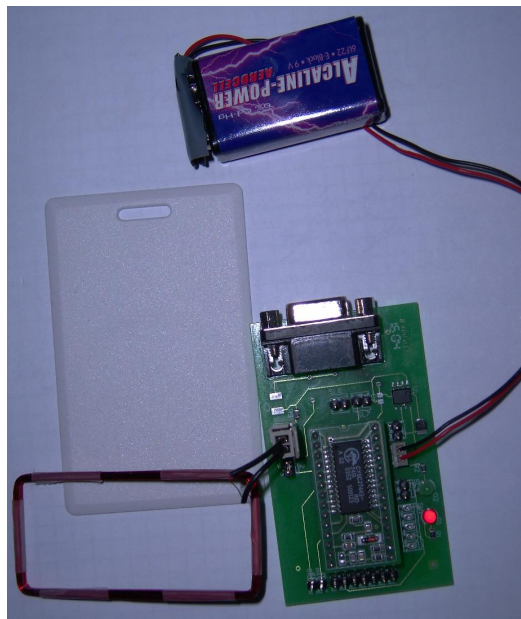


Abbildung 5.2.: SONMicro RFID-Reader

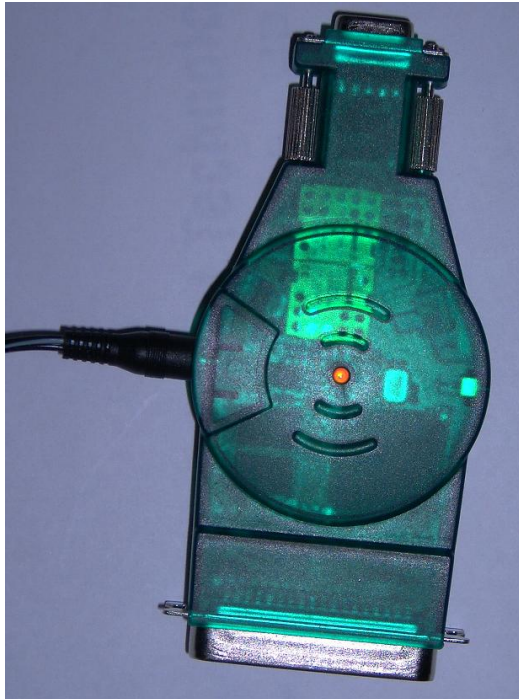


Abbildung 5.3.: Bluetoothmodul von PicoPlug

Der niedrige Anschaffungspreis spricht für das RFID-Modul, da dieses schon ab 13,90\$ zu erwerben ist. Zudem ist ein Microcontroller integriert, welcher durch entsprechende Programmierung das Modul um weitere Features erweitern kann. Für Testzwecke wird RFID das Basic Development Kit SM2005-B5 genutzt, da dieses eine Programmierung ermöglicht und eine COM-Schnittstelle mitliefert. Allerdings ist es auf Grund des hohen Preises und der Bauform des Kits² sinnvoll, ein eigenes Gerät zu entwickeln, welches das genannte Modul um eine Bluetoothschnittstelle erweitert.

Über diese Schnittstelle ist das Bluetoothmodul verbunden. Leider bietet das Modul von PicoPlug keine interne Stromversorgung, so dass der Einsatz nur für Testzwecke nutzbar ist. Außerdem ist dieses Modul recht groß, was einen mobilen Einsatz unkomfortabel machen dürfte. Besonders der enthaltene Druckerport ist für den gewünschten Einsatzbereich uninteressant.

²Ein solches Kit ist ab 40\$ zu erhalten, wobei dieser Preis erst ab 10 Einheiten angeboten wird.

5.2. Programmiersprache

Soweit es möglich war, wurde die Software in Java entwickelt, da diese Programmiersprache plattformunabhängig ist. Dadurch ist es möglich, ohne größeren Aufwand die Software auf ein anderes System zu portieren, was den flexiblen Einsatz der Software steigert. Allerdings besteht auch ein Nachteil durch die Plattformunabhängigkeit, da durch diese keine direkte Ansteuerung von Hardwareschnittstellen möglich ist. Zwar definiert Sun Microsystems diese Schnittstellenansteuerung, aber die Realisierung wird anderen Anbietern überlassen. Aus diesem Grund wurde eine freie Bibliothek genutzt, die eine solche Ansteuerung ermöglicht. Genaueres zu dieser sowie anderer genutzter Bibliotheken folgt später.

Zudem ist zu Java noch zu sagen, dass diese Programmiersprache eine einfache Handhabung für verteilte Systeme bietet. Viele Verbindungs- und Kommunikationsmöglichkeiten werden direkt von Sun Microsystems unterstützt, was natürlich auch zur Verbreitung beiträgt. Dadurch ist die Einbindung von Web-Services relativ einfach, da Sun Microsystems auch auf diese Technologie geachtet hat und sie somit unterstützt wird.

Ein weiterer Grund für die Nutzung von Java ist die weite Verbreitung. Viele Projekte werden in Java entwickelt, so dass zum Einen viele freie Bibliotheken verfügbar sind. Zum Anderen zeigt es, dass diese Sprache zukunftsorientiert und somit die Wahrscheinlichkeit höher ist, dass eine Weiterentwicklung dieser Software umgesetzt werden kann.

5.3. Clientsoftware

Die Clientsoftware soll möglichst klein gehalten werden, damit die Auswahl der Hardware wenigen Einschränkungen unterliegt. Das im Designkapitel beschriebene Klassendiagramm (siehe Abb. 4.2) findet in dieser Software seine direkte Realisierung. Die Aufteilung der einzelnen Control-Klassen wurde realisiert, um die einzelnen Kommunikationsarten übersichtlich steuern zu können.

Um unter Java eine Ansteuerung der Bluetoothkomponente zu ermöglichen, wird auf eine freie Bibliothek zurückgegriffen, welche den direkten Zugriff auf eine COM-Schnittstelle ermöglicht. Hierfür bietet sich CEJavaComm³ an, welche in der Version 0.8pre2 eingebunden wurde. Da die genutzte Bluetoothverbindung einen Com-Port simuliert, kann mit dieser Softwarebibliothek darauf zugegriffen und gesteuert werden. Zwar handelt es sich um eine Version in einem frühen Entwicklungsstadium, aber im Verlauf dieser Arbeit konnten keine fehlerhaften Funktionen der Bibliothek erkannt werden. Zudem wurde diese Bibliothek auch in anderen Arbeiten schon erfolgreich eingesetzt (siehe z.B. Stein (2005)).

³Die frei verfügbare Bibliothek ist unter <http://www.teilo.net> zu finden.

Wie schon in Kapitel 4 beschrieben soll noch einmal auf die beiden Teilaspekte der Informationsdarstellung sowie der Lesegerätidentifikation genauer eingegangen werden. Zudem werden die Kommandosatzdateien genauer betrachtet und ihre Möglichkeiten aufgezeigt. Dieses ist in den folgenden Unterkapiteln nachzulesen.

Informationsdarstellung

Es gibt zwei Schwerpunkte, die für die grafische Oberfläche zu beachten sind. Zum Einen soll diese sehr übersichtlich sein, was eine einfache Nutzung des Systems ermöglichen wird. Hierfür wurden, wie Abb. 5.4 zu entnehmen ist, zwei Buttons sowie ein Textfeld eingebunden. Da das System vorerst nur zwei Aktionen bietet können diese direkt über die entsprechenden Buttons angesprochen werden. Das erhöht die Übersichtlichkeit der Oberfläche und erleichtert die Bedienung.

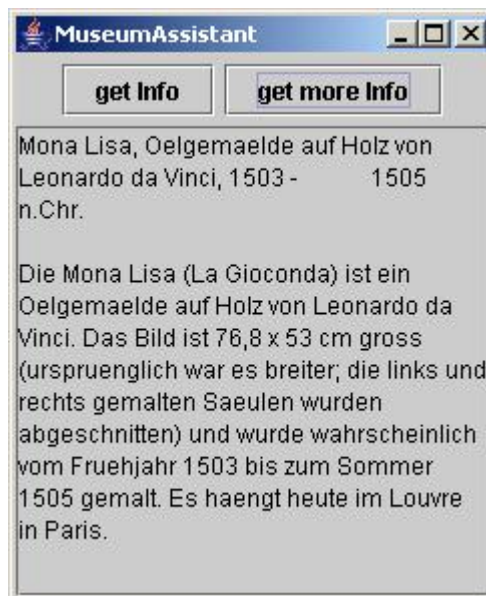


Abbildung 5.4.: MuseumAssistant

Zur Informationsdarstellung wurde ein Textfeld eingebunden, was vorerst verwundern mag. Zwar wäre es möglich, Bilder sowie Filme als weitere Informationen darzustellen, aber die Informationsdarstellung soll vorerst exemplarisch erstellt werden. Daher werden diese Informationen nicht zwingend benötigt, weswegen eine rein textliche Darstellung der Informationen gewählt wurde. Zudem bietet der PDA ein sehr begrenztes Display, was die Betrachtung von Bild- oder Filmmaterial sehr einschränkt. Da diese Informationen aber evtl. für den Nutzer von Interesse sein könnten, könnte eine Erweiterung des Systems sinnvoll sein. Auf eine entsprechende Lösung in Kapitel 7 im Ausblick etwas genauer eingegangen.

Zwar bietet das genutzte PersonalJava nur AWT für eine Oberflächenprogrammierung an, aber das genutzte System wurde um das aktuellere Swing erweitert. Dieses wurde in der Version 1.1.1 integriert, wobei ein kleiner Bug beseitigt werden musste⁴, um die fehlerfreie Nutzung zu garantieren. Dadurch ist eine komfortable und aktuelle Oberflächenprogrammierung möglich, was spätere Erweiterungen zudem vereinfachen sollte.

Lesegerätidentifikation

Aus dem Kapitel 4.3 ist zu entnehmen, dass eine automatisierte Identifikation der Leserhardware genutzt werden soll. Hierfür werden die einzelnen Kommandosätze in entsprechenden XML-Dateien beschrieben. Die dafür definierte DTD finden Sie im Anhang A.

Diese DTD wurde möglichst offen gestaltet, so dass der Großteil von verschiedenen Kommandosequenzen abgebildet werden kann. Während der Identifikation entnimmt das System der Reihe nach die einzelnen Kommandosätze aus einem entsprechenden Verzeichnis. Dabei wird die Identifikationssequenz abgearbeitet. Sobald das System eine solche Sequenz fehlerfrei abarbeiten kann, wird dieser Kommandosatz für das System bereitgestellt und die Identifikation ist damit abgeschlossen.

Somit ist es sehr wichtig, dass eine eindeutige Identifikationssequenz für jedes Lesegerät erstellt wird. Sobald zwei Geräte dieselbe Sequenz nutzen, wird das System den ersten Kommandosatz bereitstellen, was natürlich zu Problemen führen kann. Zwar steht dann dem System ein Kommandosatz zur Verfügung, aber dieser muss nicht passend zur Hardware sein. Die Sequenz zum Auslesen einer ID kann dann zu Fehlermeldungen führen. Allerdings ist zum Einen die Wahrscheinlichkeit gering, dass zwei Geräte eine identische Sequenz benötigen. Zum Anderen werden wahrscheinlich in einem Gesamtsystem nur eine Sorte von Lesegeräten genutzt, da sich dadurch Kosten minimieren lassen. Somit wird auch nur ein Kommandosatz benötigt, was Probleme dieser Art von vornherein ausschließt.

Um aus den XML-Dateien Kommandosequenzen für das System zu erstellen, müssen diese Dateien geparkt werden. Da es sich um eine relativ kleine DTD handelt, wurde ein eigener Parser realisiert. Zwar gibt es einige Parser, die für mobile Clients entwickelt wurden. Diese bieten aber wesentlich mehr Funktionsumfang, als es für diesen Fall benötigt wird. Das wird schon allein dadurch deutlich, dass ein solcher Parser allgemein gehalten ist, damit er alle Varianten einer DTD flexibel parsen kann. Durch einen eigenen Parser, der speziell auf die genutzte DTD ausgerichtet ist, lässt sich die Software relativ klein halten.

Ein Nachteil für das System ist, dass durch Veränderungen an der DTD auch die Software im Bereich des Parsers angepasst werden muss. Da aber die DTD sehr frei gestaltet wurde und somit Erweiterungen eher unwahrscheinlich sind, wird auf diese Flexibilität auf Grund

⁴Eine Anleitung dazu findet sich z.B. hier: http://www.blueboard.com/j2me/notes/2002_7_26.htm

der geringeren Softwaregröße verzichtet. Zudem muss auch mit einem allgemeinen Parser die Software angepasst werden, damit diese richtig reagiert. Somit ist der Nachteil relativ gering.

Kommandosätze

In einem Kommandosatz sind verschiedene Einstellungsparameter für die Verbindung sowie Kommandosequenzen zu finden. Zu Beginn werden die Parameter ausgelesen, was die Baudrate, Anzahl der Datenbits und der Stopbits sowie die Einstellung für die Paritätsprüfung umfasst. Daraufhin wird eine entsprechende RFCOMM-Verbindung erstellt.

Des Weiteren beinhaltet ein Kommandosatz einen Namen, welcher bei erfolgreicher Identifikation des Lesegeräts auf dem PDA angezeigt wird. Dieses wird als weitere Absicherung eingesetzt, damit der Nutzer die Möglichkeit hat zu überprüfen, ob die korrekte Hardware identifiziert wurde. Sobald die erste ID ausgelesen wird, wird dieser Name aber wieder überschrieben, da er im laufenden Betrieb für den Nutzer unwichtig ist. Außerdem lässt sich ein Inittest einsetzen, bei dem ein Tag direkt vor der Ausgabe des Clients eingelesen wird. Dabei wird ein Funktionstest gemacht und als Informationstext zur eingelesenen ID könnte man einen Willkommenstext ausgeben.

Der Zusammenhang zwischen Kommandosatz, Kommandosequenz sowie Kommando wurde schon in Kapitel Design erläutert. Eine Kommandosequenz beinhaltet einen Namen, der vorerst für spätere Erweiterungen gedacht ist. Durch diesen Parameter sollten verschiedene Kommandosequenzen unterscheidbar sein. Zudem ist in einer Kommandosequenz eine Reihe von Kommandos enthalten, wobei zwischen „commandosend“ und „commandorec“ unterschieden wird. Abschließend kann noch angegeben werden, wie viele Bytes am Ende einer Kommandosequenz empfangen werden sollen. Dieses wird zur Übertragung der ID genutzt, da diese keine allgemein festgelegte Länge aufweist.

Nun wird der Aufbau eines Kommandos betrachtet. Wie erwähnt gibt es zwei Arten von Kommandos. „commandosend“ entspricht einem Sendekommando und „commandorec“ einem Empfangskommando, wobei dieses aus der Sicht des PDAs zu verstehen ist. Dadurch entscheidet das System, wie das entsprechende Kommando zu verarbeiten ist. Ansonsten besteht ein Kommando, egal von welcher Sorte, von mindestens einem Parameter.

Ein Parameter kann verschiedene Informationstypen beinhalten, welche das Erstellen eines Kommandosatzes erleichtern soll. Beim Typ „byte“ wird bis zu einem Byte binär angegeben. Falls weniger als acht Ziffern angegeben werden, füllt das System das Byte vorne mit Nullen auf. Dieses wurde eingebracht, um eine fehlerträchtige Umrechnung von binär in hexadezimal zu vermeiden.

Der Typ „int“ ist ansich für zu empfangene Kommandos gedacht. Mit diesem Typ gibt man an, wie viele Bytes bei Abarbeitung des empfangenen Kommandos ignoriert werden können. Da empfangene Kommandos mit den vorgegebenen aus der Kommandosequenz verglichen werden, ist es somit möglich eine Anzahl von Bytes beim Vergleich zu überspringen. Falls dieser Typ bei einem zu sendenden Kommando genutzt wird, so wird dieser genauso wie der Typ „string“ zeichenweise versendet.

Abschließend gibt es noch den Typ „cwlc“. Diese Abkürzung steht für „compare with last commandtosend“. Jeweils das zuletzt gesendete Kommando wird im System zwischengespeichert. Falls ein Kommando durch Wiederholung bestätigt werden soll, nutzt man diesen Typen, der allerdings keine Informationen beinhaltet. Es ist zwar möglich, diesem einen Inhalt zu geben, aber das System wird ihn nicht beachten.

Als Beispiel folgt ein Auszug aus dem Kommandosatz des genutzten RFID-Lesers. Im nachfolgenden Sequenzdiagramm (siehe Abb. 5.5) ist die entsprechende Kommunikation dargestellt. Dabei ist zu beachten, dass durch „ gekennzeichnete Zeichen im Klartext versendet werden. Sobald eine Kennzeichnung durch ‚ zu sehen ist, handelt es sich um Hexadezimalwerte, die jeweils ein byte darstellen. Abschließend deutet ID darauf hin, dass die ausgelesene ID versendet wird, die in diesem Beispiel 56 bytes lang ist.

```
<commandtosend>
<parameter type="string">re064</parameter>
<parameter type="byte">10</parameter>
<parameter type="byte">1</parameter>
<parameter type="byte">111</parameter>
</commandtosend>
<commandtorec>
<parameter type="cwlc"></parameter>
</commandtorec>
<commandtosend>
<parameter type="string">acknwlge</parameter>
</commandtosend>
<commandtorec>
<parameter type="string">acknwlge</parameter>
</commandtorec>
<commandtorec>
<parameter type="string">acknokok</parameter>
</commandtorec>
<numberofreceivingbytes>56</numberofreceivingbytes>
```

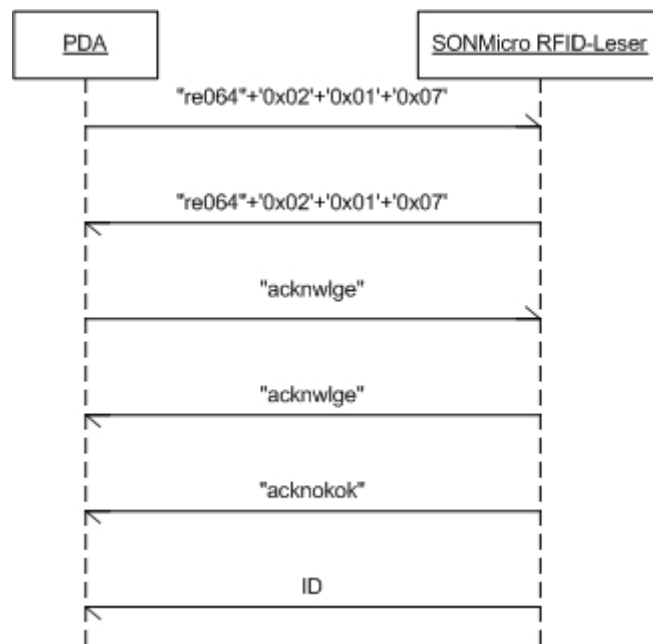


Abbildung 5.5.: Beispiel Kommunikationsablauf

5.4. Serversoftware

Für die Serverseite des MuseumAssistant wird ein SOAP-Dienst benötigt. Dieser wurde in Java realisiert, wobei als Rahmensystem für den Dienst ein Tomcat als Webserver dient. Die Auswahl fiel auf diesen, da bei einem Tomcat die Erweiterung durch SOAP-Dienste relativ problemlos möglich ist. Die entsprechenden Pakete von Sun Microsystems müssen nur eingebunden und die jeweiligen Schnittstellen angemeldet werden. Nach dem Start des Webservers ist dann die Nutzung des Dienstes möglich.

Da aus dem Design deutlich wird, dass sowohl ein Web-Service sowie eine Datenbank für die Realisierung benötigt werden, wird in den beiden folgenden Abschnitten genauer auf diese Bereiche eingegangen. Dabei erfolgt eine Beschreibung der Realisierung und die Nutzung der entsprechenden Bibliotheken wird begründet. Zur Verdeutlichung beachten Sie bitte die Abbildung 4.4.

Web-Service

Wie schon zu Beginn des Unterkapitels zur Realisierung der Serversoftware erwähnt, ist ein Tomcat als Webserver leicht um einen SOAP-Dienst zu erweitern. Nachdem mehrere Pakete dem Tomcat bereitgestellt und Umgebungsvariablen angepasst wurden, sind bereits

alle Rahmenbedingung für den Server erfüllt, so dass dieser SOAP-Dienste anbieten kann⁵. Die benötigten Pakete bietet Sun Microsystems direkt an, wobei die Nutzung kostenlos ist. Dieser Punkt sowie die flexible Nutzung spricht besonders für SOAP.

Mit WSDL beschreibt man nun die entsprechenden Schnittstellen. Diese Beschreibung wird beim Anmelden des Dienstes angegeben, wodurch der Server die benötigten Einstellungen übernimmt. Nun muss die Funktionalität programmiert werden, wobei diese später durch Austausch der entsprechenden Dateien erneuert werden kann. Solange die Schnittstellen gleich bleiben, wird keine Neuanschaffung benötigt. Einzig ist zu beachten, dass für ein solches Update der Server neu gestartet werden muss.

Datenbank

Ausstellungsobjekte werden durch die Klasse ArtObjekt dargestellt. Objekte dieser Klasse sollen in einer Datenbank hinterlegt werden, damit diese leicht wartbar und im Betrieb verfügbar sind. Um die Wartbarkeit zu verbessern, werden alle Objekte in Listen in entsprechenden XML-Dateien hinterlegt. Der Aufbau dieser ist in einer DTD beschrieben (siehe Anhang B).

Dadurch ist es möglich, neue Objekte durch eine solche Datei zu beschreiben und diese in das entsprechende Unterverzeichnis zu kopieren. Beim nächsten Start des Webservers durchsucht das System dieses Verzeichnis und parst alle entsprechenden XML-Dateien. Die dabei eingelesenen Objekte werden in einer Datenbank hinterlegt, welche dann dem System diese zur Verfügung stellt.

Wie schon im Kapitel Design beschrieben gibt es nur minimale Einschränkungen an die Datenbank. Da ein Museum regelmäßig geschlossen ist, können alle Wartungsarbeiten sowie Updates ohne direkte Ausfallzeiten erledigt werden. Daher muss die Datenbank keine Wartungsarbeiten im laufenden Betrieb ermöglichen. Zudem werden alle Informationen über eine eindeutige ID abgerufen, was zu einer einfachen Struktur der zu verwaltenden Daten führt. Im laufenden Betrieb werden überwiegend Datenbankabrufe erwartet, weswegen eine hohe Performance dafür zu bieten ist.

Durch diese minimalen Anforderungen bietet sich die Datenbank db4O an. Diese ist frei verfügbar und steht unter der GPL, so dass die Nutzung und sogar evtl. benötigte Anpassungen möglich sind. Zudem ist sie als Javabibliothek erhältlich, was die Einarbeitung erleichtert und Kompatibilitätsprobleme ausschließt. Allerdings wurde die Performance für Abfragen noch nicht getestet. Dieses ist aber für das Grundgerüst erstmal weniger wichtig.

⁵Eine Anleitung findet man z.B. hier: <http://www.onjava.com/pub/a/onjava/2002/02/27/tomcat.html>

5.5. Kommunikation

Da zwei unterschiedliche Arten der Kommunikation in dieser Arbeit genutzt werden, werden die beiden hier in ihrer Realisierung veranschaulicht. Der prinzipielle Ablauf einer solchen Kommunikation wurde schon in Kapitel 4.4 beschrieben. Hier wird auf die konkrete Realisierung eingegangen, wobei eine detaillierte Ansicht deutlich wird.

PDA - Lesegerät

Über Bluetooth wird per RFCOMM eine virtuelle COM-Schnittstelle simuliert, welche vom PDA aus angesteuert wird. Diese wird durch die oben erwähnte Bibliothek CEJavaComm angesteuert, wodurch die Kommandosequenzen abgearbeitet werden können. Nachdem diese Sequenzen dem System verfügbar gemacht wurden, tritt die Klasse BlueControl in Aktion, welche die komplette Kommunikation per Bluetooth kontrolliert. Dabei wird eine Übertragungsrate von 38400 Bd/s genutzt, da höhere nicht von allen Komponenten unterstützt werden.

Nachdem alle benötigten Parameter der Kommandosatzdatei entnommen wurden, wird eine Verbindung zur Gegenstelle, dem Lesegerät, aufgebaut. Hierbei ist zu beachten, dass diese auch auf die gewählten Parameter eingestellt sind, da sonst eine Kommunikation unmöglich wird. Der Ablauf der Kommandosequenz wird auch der Datei entnommen, so dass das System diesen direkt befolgt. Somit werden die benötigten Kommandos an die Klasse BlueControl geben, welche dann die Daten versendet bzw. empfängt. Das Abgleichen von Daten wird in der Klasse ReaderControl gemacht. Dadurch werden hier entsprechende Fehler erkannt und das System reagiert auf diese entsprechend.

Diese Aufteilung vereinfacht die Wartung des Sourcecodes. Alle Änderungen, die sich auf die Kommandosätze beziehen, sind durch Anpassungen in der Klasse ReaderControl zu erreichen. Des Weiteren kann die Bluetoothverbindung durch Bearbeitung der Klasse BlueControl verändert und den Umständen angepasst werden.

Zudem soll an dieser Stelle darauf hingewiesen werden, dass der konkrete Ablauf der Kommunikation vom genutzten Kommandosatz abhängig ist. Somit ist es auch möglich, dass mehrere Kommandos zuerst vom PDA an das Lesegerät gesendet werden, bevor dieses dann entsprechend eine Rückmeldung schickt. Theoretisch wäre auch eine unidirektionale Verbindung möglich, wobei dieses für die vorgegebene Aufgabenstellung wenig Sinn macht. Schließlich soll eine ID ausgelesen und verschickt werden.

PDA - Server

Der Server bietet seine Dienste per SOAP an, da diese Technik einen flexiblen Einsatz ermöglicht. Im Testszenario wird HTTP für die Datenübertragung genutzt, da sich dieses durch die bestehende WLAN-Verbindung anbietet. Hierfür wird allerdings die IP des Servers benötigt, welche direkt im Sourcecode in der Klasse ServerControl zu finden ist.

Als Alternative zu der hart codierten IP wäre ein Domainname nutzbar, wobei dafür von jedem Client ein DNS-Server erreichbar sein müsste. Dadurch lässt sich eine flexiblere Einstellung des Servers erreichen. Da aber ein solcher Server nicht Teil der Testumgebung ist, wurde auf diese Variante verzichtet. Zudem wird das System in einem lokalen Netzwerk genutzt, so dass ein DNS-Server nicht zwingend erreichbar ist. Die Erweiterung lässt sich aber durch minimaler Veränderung am Sourcecode erreichen.

Zur Abfrage des Servers wird auf dem PDA eine entsprechende SOAP-Nachricht erstellt, welche dann an die IP des Server geschickt wird. Dieser wertet die empfangene Nachricht aus und reagiert darauf, wobei er seine Ergebnisse wiederum in einer SOAP-Nachricht zurücksendet. Der Client entnimmt dieser die angeforderten Informationen und stellt sie dem Nutzer dar.

5.6. Zusammenfassung

Dieses Kapitel beschreibt die Realisierung des Gesamtsystems. Nach der Beschreibung der genutzten Hardware wird dem Leser erklärt, warum Java als Programmiersprache gewählt wurde. Darauf folgt sowohl die Beschreibung der realisierten Clientsoftware als auch der Serversoftware. Zum Abschluss wird beschrieben, wie die einzelnen Kommunikationen eingesetzt werden.

6. Systemtest

Nach der Realisierung des System folgen Funktionstests. Diese dienen dazu, sowohl die Funktionsfähigkeit der einzelnen Komponenten als auch des Gesamtsystems zu überprüfen. In diesem Kapitel werden die einzelnen Bereiche der Hardware sowie der Kommunikation betrachtet und getestet. Darauf folgt ein Test der Systemsoftware, womit die Funktionsfähigkeit von MuseumAssistant aufgezeigt werden soll.

6.1. Hardware

Es werden verschiedene Hardwarekomponenten benutzt, auf die in diesem Test näher eingegangen wird. Es wird ein iPAQ H5550 genutzt, welcher die Systemumgebung der Clientsoftware bietet. Neben dem 400-MHz-XScale-Prozessor bietet dieser PDA sowohl 128 MB RAM sowie 48 MB ROM, was genügend Performance für einen sinnvollen Einsatz als Client bereitstellt. Zudem ist durch den Einsatz von SD-Karten eine Erweiterung des Speicherplatzes möglich. Dieses wird aber für den Einsatz von MuseumAssistant nicht benötigt.

Leider benötigt die Software ca. 30 Sekunden bis das System einsatzbereit ist. Da dieses aber auch die Erstellung der Bluetoothverbindung und Identifikation der Hardware beinhaltet, ist dieser Wert akzeptabel. Im laufenden Betrieb ist die Reaktionsfähigkeit nur von der Kommunikation abhängig.

Der H5550 bietet einen Lithium-Ionen-Akku mit 1.250 mAh. Zwar ist dieses allgemein für einen PDA mehr als ausreichend, aber für MuseumAssistant entstehen dadurch kleine Einschränkungen. Durch die Tatsache, dass ein Client während eines ganzen Museumsbesuch aktiv genutzt werden soll, dürfte diese Akkuleistung zwar dafür ausreichend sein. Allerdings soll ein Client über den gesamten Tag eingesetzt werden, was nur durch einen Austausch des Akkus zwischen den Besucherwechseln erreicht werden kann.

Des Weiteren wird ein RFID-Lesegerät genutzt, um die Ausstellungsstücke zu identifizieren. Im Kapitel Realisierung wurde beschrieben, dass für Testzwecke die Kombination von einem Bluetoothmodul der Firma PicoPlug und einem RFID-Leser der Firma SONMicro eingesetzt wird. Dadurch wird deutlich, dass diese Hardware nur für den Testeinsatz sinnvoll nutzbar ist, wobei sie hier ihre Aufgaben gut erfüllt.

Allerdings soll an dieser Stelle darauf hingewiesen werden, dass sowohl das Datenblatt als auch das Handbuch zum RFID-Leser zum Zeitpunkt dieser Diplomarbeit fehlerhaft sind. Es empfiehlt sich daher, einen Portlogger während der Entwicklung zu nutzen, um derartige Fehler möglichst frühzeitig zu erkennen. Die Firma SONMicro wurde auf die Fehler hingewiesen, wobei eine Verbesserung angekündigt wurde¹.

Ansich ließe sich durch eine Erweiterung des Bluetoothmoduls um ein Akku die Mobilität des Ganzen erreichen. Zum Einen ist es für die Testanwendung nicht notwendig, da hier die Kommunikation zwischen den Komponenten sowie der Systemablauf in den Vordergrund gestellt werden. Zum Anderen wäre die Entwicklung eines kompakteren RFID-Lesers sinnvoller, welchem ein Bluetoothmodul integriert wäre. Da das Lesegerät als Microcontroller vorliegt, dürfte eine solche Entwicklung mit relativ geringem Aufwand möglich sein. In dieser Arbeit liegt der Schwerpunkt aber nicht auf diesem Bereich, weswegen darauf verzichtet wurde.

In der Testumgebung wird des Lesemodul von SONMicro auf der Programmierplatine eingesetzt. Diese bietet Anschlüsse für Batterie und Antenne sowie einen COM-Port. Allerdings benötigt dieses Modul mit Programmierplatine relativ viel Strom, was im späteren Einsatzgebiet zu Problemen führen könnte. Ein großer Nachteil bei der Nutzung mit einem schwachen Akku ist, dass das Gerät zwar vorerst mit richtigen Kommandos reagiert, aber dann keine ID mehr liefert. Wahrscheinlich liegt dabei eine zu geringe Spannung für die Übertragung an das Tags an, so dass dieses nicht die benötigte Energie für eine Antwort gewinnen kann. Eine detaillierte Untersuchung dieses Problems wurde allerdings nicht durchgeführt.

Zudem bietet dieses Modul mit den entsprechenden RFID-Tags nur eine Lesereichweite von ca. 12cm. Das ist für den späteren Einsatz zu wenig, da die Mobilität des Besuchers dadurch sehr stark eingeschränkt wird. Somit wird zu einer entsprechenden Auswahl eines Lesegeräts geraten, wobei eine Lesereichweite von mindestens 50cm möglich sein sollte.

Natürlich darf diese Reichweite auch nicht zu groß werden, da sonst mehrere Tags im Aktionsradius sind und somit kein eindeutiges Objekt identifiziert werden kann. Hierfür wäre eine spätere Erweiterung des Systems denkbar, wobei der Nutzer dann eine Auswahl treffen muss, um entsprechende Informationen abzurufen.

6.2. Kommunikation

Für die Kommunikation zwischen PDA und RFID-Leser wird eine Bluetoothverbindung benötigt. Diese muss von Hand eingestellt werden, was vor der Ausgabe des Clients gemacht werden sollte. Dadurch wird verhindert, dass solche Details Nutzer abschrecken könnte. In

¹Die Fehler sind zur Zeit (11.06.2005) noch immer enthalten.

der Testumgebung gibt es eine überschaubare Auswahl von Bluetoothmodulen, aber beim Einsatz in einem Museum dürfte das anders aussehen. Da dort eine Reihe von PDAs sowie Lesegeräte mit Bluetooth vorhanden sind, könnte die Auswahl verwirrend sein. Hinzu kommen bluetoothfähige Geräte, die von Besuchern mitgeführt werden².

Die Einstellungen in der Software laufen nach der Auswahl des Moduls automatisch ab. Allerdings müssen die Hardwaremodule auf die passenden Einstellungen konfiguriert werden. Dieses wird zwar im Allgemeinen nur einmal notwendig sein, da die Einstellungen gespeichert werden. Aber diese Notwendigkeit birgt eine Fehlerquelle, da eine Anpassung der Einstellungen an drei Stellen unternommen werden muss³. Allerdings sollte eine solche Umstellung durch entsprechendes Fachpersonal durchgeführt werden, so dass dieses Problem bekannt sein sollte.

Da die Baudrate frei gewählt werden kann⁴, musste in der Software darauf durch relativ hohe Timeoutzeiten reagiert werden. Dadurch können auch geringe Übertragungsraten genutzt werden. Allerdings haben diese hohen Zeiten den Nachteil, dass die Kommunikation evtl. gebremst wird, da unter Umständen relativ lange auf eine Rückmeldung gewartet wird. Dafür werden Fehlerquellen vermieden, da keine Kommandos durch zu geringe Wartezeiten verloren gehen.

Für die Serverabfragen wird eine WLAN-Verbindung genutzt. Hierbei treten im Testsystem keine Probleme auf. Zwar wird eine Ad-hoc-Verbindung genutzt, da im Testsystem kein Access Point zur Verfügung steht. Aber diese Verbindung muss zu Beginn einmal angewählt werden und die Kommunikation zwischen den beiden Systemkomponenten funktioniert. Im Endbetrieb soll die WLAN-Anbindung über entsprechende Access Point laufen. Dabei werden entsprechende Einstellungen gespeichert, so dass diese nicht vor dem Programmstart erledigt werden müssen.

Zudem wurde auf verschiedene Sicherheitsmechanismen wie z.B. Verschlüsselung verzichtet, da keine sicherheitsrelevanten Daten übertragen werden. Falls im späteren System dafür Bedarf besteht, können entsprechende Einstellungen vorgenommen werden, ohne dass MuseumAssistent angepasst werden muss.

Die im Abschnitt 2.4 beschriebenen möglichen Probleme konnten im Testsystem nicht überprüft werden. Da nur ein Client zur Verfügung steht und bei diesem durch die Vorgaben der Software nie beide Techniken parallel eingesetzt werden, war ein Test nicht möglich. Selbst bei einem erzwungenen Paralleleinsatz sind keine merkbaren Unterschiede festzustellen,

²Aktuelle Handys verfügen z.B. häufig über diese Schnittstelle. Zudem ist Bluetooth weit verbreitet, wodurch es eine Vielzahl von anderen Geräten mit dieser Schnittstelle gibt.

³Neben den Anpassungen im Kommandosatz müssen sowohl das Bluetoothmodul des PDAs als auch das des Lesegeräts entsprechend eingestellt werden.

⁴Die genutzten Bluetoothmodule müssen allerdings die gewählten Einstellungen unterstützen.

da dafür das Datenaufkommen zu gering ist. Somit ist ein gegenseitiges Ausbremsen nicht festzustellen.

Trotzdem sollte dieser Test vor dem Einsatz in der im Szenario beschriebenen Umgebung vollzogen werden. Hierbei sind mehrere Clients zu betreiben, so dass ein entsprechendes Datenaufkommen entsteht. Dabei könnte sich herausstellen, dass diese Verzögerungen in einem akzeptablen Rahmen bleiben, weswegen das System mit den günstigeren Kommunikationskomponenten einsetzbar ist. Um einen solchen Test zu umgehen, empfiehlt sich, die im entsprechenden Abschnitt erwähnte Hardware einzusetzen, da damit diese Wechselwirkungen nicht eintreten.

Abschließend ist noch zu sagen, dass ein Test mit mehreren aktiven Lesegeräten gemacht werden sollte. Es ist dabei mit Störungen zu rechnen, da für diese Situation bisher kein definiertes Protokoll zur Kollisionsvermeidung realisiert wurde. Diese Problematik muss vor dem Endbetrieb getestet und angepasst werden.

6.3. MuseumAssistant

Da nun im Test sowohl auf die Hardware als auch auf die Kommunikation zwischen den Systemkomponenten eingegangen wurde, folgt eine Testbeschreibung des Gesamtsystems. Um allgemein MuseumAssistant zu testen, wird der Systemablauf für einzelne Anwendungsfälle beschrieben. Zuerst wird dabei auf den Start des Systems eingegangen. Darauf folgen die Abfragen von allgemeinen sowie erweiterten Informationen. Zum Abschluss wird noch eine unbekannte ID eingelesen und das Ergebnis aufgezeigt.

Systemstart

Zuerst sollte überprüft werden, ob der Server in Betrieb ist. Danach steht eine Überprüfung der WLAN-Verbindung an. Diese beiden Schritte können zwar auch im laufenden Betrieb gemacht werden, aber da der Client von einem Besucher genutzt wird, sollten möglichst alle Fehlerquellen vorher ausgeschlossen werden.

Kurz nachdem die Clientsoftware gestartet wurde, erscheint ein Auswahlmenü mit Bluetoothmodulen, die eine Verbindungsmöglichkeit bieten. Durch die Auswahl des entsprechenden Moduls wird eine Verbindung aufgebaut und der Identifikationsmechanismus für das Lesegerät gestartet. Nachdem dieser die Hardware erkannt hat, wird der Name des Lesegeräts wie in Abb. 6.1 ausgegeben. Sobald dieser Zustand erreicht ist, ist das System einsatzbereit.



Abbildung 6.1.: MuseumAssistant

Abrufen allgemeiner Informationen

Zum Auslesen allgemeiner Informationen muss das Lesegerät in Reichweite des entsprechenden RFID-Tags sein. Durch Betätigen des „get Info“-Buttons wird die Kommunikation zwischen PDA und Identifikationshardware gestartet. Hierbei wird die ID des Tags ausgelesen und an den PDA übertragen.

Danach startet dieser eine Serverabfrage per SOAP, wobei die ausgelesene ID mit übertragen wird. Sobald die entsprechende Rückmeldung empfangen und ausgewertet ist, wird der bisherige Inhalt des Ausgabefeldes gelöscht. Die empfangenen allgemeinen Informationen werden dann dem Benutzer dargestellt. Eine Beispielausgabe sehen Sie in Abb. 6.2.

Abrufen erweiterter Informationen

Bei Interesse kann der Nutzer weitere Informationen zum ausgewählten Ausstellungsobjekt abrufen. Dazu nutzt dieser den „get more Info“-Button. Da sich erweiterte Informationen immer auf das Objekt beziehen, zu dem zuletzt allgemeine Informationen abgerufen wurden, wird keine neue ID eingelesen. Das System speichert die letzte ID zwischen und ruft sie bei dieser Abfrage ab.

Um eine Ausgabe wie in Abb. 6.3 zu erstellen, sendet der PDA eine SOAP-Nachricht an den Server. Dieser schickt eine entsprechende Rückmeldung, die die erweiterten Informationen



Abbildung 6.2.: Anzeige allgemeiner Informationen



Abbildung 6.3.: Anzeige erweiterter Informationen

enthält. Nachdem diese aus der Nachricht ausgelesen wurden, werden sie auf dem Ausgabefeld ausgegeben. Dabei wird nach einer Leerzeile der Text angehängt. Somit hat der Nutzer sowohl die allgemeinen als auch die erweiterten Informationen auf dem Display. Das sollte beim Erstellen der Informationsdatenbank berücksichtigt werden, damit keine Informationen unnötig doppelt ausgegeben werden.

Nutzung unbekannter ID

Abschließend folgt ein Test mit einer dem Server unbekanntem ID. Dieses sollte in der Endanwendung zwar nicht vorkommen, aber das System erkennt unbekannte ID-Abfragen und gibt eine entsprechende Meldung aus (siehe Abb. 6.4).



Abbildung 6.4.: Anzeige bei unbekannter ID

In der aktuellen Version wird nur dem Nutzer eine solche Mitteilung ausgegeben. Es sind aber Erweiterungen denkbar, auf die im letzten Kapitel eingegangen werden. Schließlich dürfte es für den Betreiber von Interesse sein, dass nicht alle IDs in der Datenbank enthalten sind.

Die abgebildete Meldung ist als Beispiel anzusehen. Um Anpassungen an die Umgebungsvorgaben zu erreichen, muss im Serverdienst in der Klasse DbControl der Text entsprechend angepasst werden. Damit ist es möglich, dem Nutzer direkte Anweisungen zu geben, so dass sich dieser z.B. an das Museumspersonal wendet.

6.4. Entwicklungsstand

Im Vordergrund dieser Diplomarbeit steht das Grundgerüst des Gesamtsystems. Da sehr viele verschiedene Erweiterungen denkbar sind, die allerdings von den Gegebenheiten des Umfeldes abhängen, wurde der Schwerpunkt der Entwicklung sowohl auf ein stabiles Grundgerüst als auch auf die einfache Erweiterbarkeit von MuseumAssistant gelegt. In diesem Abschnitt soll der aktuelle Entwicklungsstand dargestellt werden. Wobei zuerst das Gesamtsystem genauer betrachtet wird. Danach folgt die Betrachtung der Software des Clients und des Servers.

Gesamtsystem

MuseumAssistant befindet sich in einem funktionsfähigen Zustand. Mit einem Client ist es möglich, per WLAN Informationen von einem Server abzurufen. Hierbei wird die Unterteilung von allgemeinen und erweiterten Informationen getroffen. Dabei wird eine ID zum Server übertragen, welcher dann eine Rückmeldung mit dem entsprechenden Text aus der Datenbank antwortet.

Dieses Grundgerüst läuft stabil und wurde sowohl mit einem PDA per WLAN als auch von einem Computer aus über ein kabelgebundenes Netzwerk getestet. Auf Grund der beschränkten Testumgebung wurden allerdings keine Lasttests durchgeführt, so dass dieses in einem entsprechenden Umfeld nachgeholt werden sollte. Da aber in der aktuellen Version nur Textinformationen versendet werden, sind Engpässe in der Bandbreite eher unwahrscheinlich.

Leider besteht in der hier besprochenen Version von MuseumAssistant ein Fehler in der Bluetoothverbindung. Die von der Clientsoftware initiierte Verbindung ist sehr unzuverlässig. Oft wird bei der Identifikation keine Daten mehr empfangen. Nachdem dieser Bluetoothkanal unterbrochen und von Hand neu aufgesetzt wird, läuft die Kommunikation fehlerfrei.

Daher wurde der Mechanismus dafür angepasst, wodurch immer der zuletzt getestete Kommandosatz dem System bereitgestellt wird. So ist zwar die Identifikation nicht mehr eindeutig, aber das System bleibt nutzbar, wenn der benötigte Kommandosatz in der letzten zu testenden Datei steht.

Durch unterschiedlichste Tests sowohl an der Software als auch an der Hardware konnte bis jetzt dieses Problem nicht gelöst werden. Aber dabei konnte das Problemfeld eingegrenzt werden, so dass die Fehlerquelle wahrscheinlich im Bluetooth-Protokollstack des iPAQs zu finden ist. Diese Vermutung wird durch die Tatsache unterstützt, dass auch in anderen Projekten mit iPAQs ähnliche Probleme mit Bluetoothverbindungen aufgetreten sind.

Clientsoftware

Die Clientsoftware setzt auch PersonalJava auf, wodurch sie für verschiedene Systeme einsetzbar ist. Auch diese Komponente des Systems läuft stabil und bietet dem Nutzer eine grafische Schnittstelle zur Informationsabfrage. Wie auf den vorherigen Abbildungen zu sehen (z.B. Abb. 6.2), kann der Nutzer über zwei Buttons die verschiedenen Aktionen abrufen.

Zur Darstellung der Informationen wird ein Textfeld genutzt. Da in dieser Version auf rein textliche Informationen zugegriffen wird, liegt die Nutzung nahe. Multimediale Informationsquellen werden bisher nicht unterstützt, da vorerst ein stabiles Grundgerüst entwickelt werden soll.

Die Identifikation der Leserhardware funktioniert. Allerdings waren bisher keine ausgiebigen Tests mit unterschiedlichen Hardwarekomponenten möglich. Daher sollte vor einem Einsatz für den Endnutzer dieser Test nachgeholt werden. Dabei dürften aber kaum Probleme auftreten, da bei einem Einsatz in einem Museum wahrscheinlich nur eine Art von Lesegeräten verwendet wird. Somit muss nur ein gültiger Kommandosatz erstellt werden, so dass das System entsprechende Steuerkommandos erhält.

Serversoftware

Da die Serversoftware einen Dienst für einen Webserver darstellt, ist die Entwicklung in diesem Bereich überschaubar. Beim Start des Dienstes werden Informationsdaten richtig geparkt und dem System bereitgestellt. Diese werden auch entsprechend versendet, wie man es auch Abb. 6.3 entnehmen kann.

Ein Lasttest für diesen Dienst steht leider auch noch aus. Allerdings kann einem Überwachungstool für die genutzte WLAN-Karte entnommen werden, dass die Auslastung nur durch einzelne Peaks auf über 1,5% geht. Somit ist bisher ein Bottleneck dort nicht zu erwarten.

Da beim Start ein Verzeichnis nach entsprechenden XML-Dateien durchsucht wird und diese dann nacheinander geparkt werden, entsteht eine gewisse Wartezeit. Aber im laufenden Betrieb sind die Daten bereits in einer Datenbank enthalten, so dass sich dieses Problem dann nicht mehr stellt.

6.5. Zusammenfassung

Dieses Kapitel befasst sich ausschließlich mit dem Test des entwickelten Systems. Nachdem genauer auf die Vor- und Nachteile der genutzten Hardware eingegangen wurde, werden die

einzelnen Kommunikationsarten getestet. Dem nachfolgenden Abschnitt ist der Programmablauf während der Nutzung von MuseumAssistant zu entnehmen. Hierdurch wird dem Leser deutlich gemacht, in welchem Zustand das System ist und welche Funktionen nutzbar sind. Dieser Aspekt wird im letzten Abschnitt genauer betrachtet. Dabei wird auf den Entwicklungsstand eingegangen, wobei sowohl das Gesamtsystem als auch die einzelnen Komponenten genauer betrachtet werden.

7. Fazit und Ausblick

Abschließend soll diese Arbeit bewertet und zudem einen Ausblick gegeben werden. Dabei werden kritisch die Arbeitsschritte sowie Ergebnisse betrachtet. Im letzten Abschnitt wird sich mit der Zukunft von MuseumAssistant beschäftigt. Hier werden Möglichkeiten der Weiterentwicklung und Verbesserungswünsche aufgezeigt.

7.1. Fazit

Auf Grund der Studienrichtung der technischen Informatik des Autors, war ursprünglich ein anderer Schwerpunkt für diese Arbeit erdacht. Es sollte eine Ansteuerung eines RFID-Lesers von einem PDA aus realisiert werden. Dadurch sollte eine mobile Identifikationsmöglichkeit erreicht werden. Da aber das verfügbare Lesegerät der Firma SONMicro eine Unterstützung der RS232-Schnittstelle bietet, war eine Erweiterung um eine mobile Ansteuerung einfach zu realisieren.

Daher wurde der Schwerpunkt auf das Grundgerüst samt Serverabfrage gelegt. Ursprünglich war das als mögliche Erweiterung nach der Diplomarbeit gedacht, aber durch die neuen Rahmenbedingungen stand einer Realisierung nichts im Weg. Somit musste sich der Autor mit einem Teil der Informatik auseinandersetzen, der von ihm bisher kaum beachtet wurde.

Durch Web-Services sind viele neue Einsatzgebiete möglich geworden, die flexible Endsysteme benötigen. Daher war es für den Autor besonders wichtig, MuseumAssistant möglichst modular zu realisieren. Durch die gekapselten Systemkomponenten ist dieses auch in einem gewissen Rahmen gelungen, so dass spätere Anpassungen an Umgebungseinschränkungen relativ leicht realisierbar sein sollten.

Leider war ein ausgiebiger Test des Identifikationsmechanismus nicht möglich. Durch fehlende Alternativgeräte waren nur vereinzelte Tests durchführbar, die über eine simulierte Gegenstellen am Computer gemacht wurden. Allerdings bietet die direkte Ansteuerung von Hardware meist andere Ergebnisse. Dieses wurde dem Autor im Laufe dieser Arbeit auch durch erhebliche Mängel sowohl im Datenblatt als auch im Handbuch des genutzten RFID-Lesers besonders deutlich.

Trotzdem ist durch diese Diplomarbeit ein Prototyp entstanden, der den angegebenen Anforderungen gerecht wird. Neben der automatisierten Identifikation des Lesegeräts wurde auch die Ansteuerung desselben realisiert. Des Weiteren ist eine Serverabfrage möglich, die dem Nutzer Textinformationen anbietet. Diese Informationen sind in einer Datenbank im Server hinterlegt, so dass auch eine zentrale Wartung der Daten realisiert wurde. Außerdem ist MuseumAssistant absolut modular aufgebaut, weswegen einem Austausch einzelner Komponenten nichts im Wege steht.

7.2. Ausblick

Gerade der modulare Aufbau ermöglicht viele verschiedene Einsatzgebiete. Daher wird zuerst auf mögliche Erweiterungen für den Einsatz im Museum eingegangen. Danach folgt eine Beschreibung verschiedener Szenarien, bei denen dieses System dem Nutzer einen hilfreichen Dienst bieten kann.

Multimedia

Da es sich bei der entwickelten Version von MuseumAssistant um einen Prototypen handelt, sind dort noch verschiedene Erweiterungen möglich und sinnvoll. Neben einer Sprachauswahl wäre auch eine Unterstützung der Informationsdarstellung durch Multimediainhalte denkbar. Dabei könnten verschiedene Sprachausgaben sowie Filme oder Animationen dargestellt werden, um dem Besucher die Informationen näher zu bringen.

Des Weiteren wäre über das vorhandene WLAN eine Internetanbindung möglich. Dadurch könnte der Besucher noch genauer über eine Suchmaschine für ihn interessante Informationen abrufen. Allerdings müssten für diese Erweiterung entsprechende Sicherheitsmaßnahmen erstellt werden, damit die Internetanbindung nur zur Unterstützung des Museumsbesuchs genutzt werden kann.

RFID-Leser

Der genutzte RFID-Leser der Firma SONMicro ist für den Einsatz in MuseumAssistant eher ungeeignet. Besonders die geringe Reichweite stört, da so der Besucher doch noch sehr an die Ausstellungsstücke und den daran zu findenden Tags gebunden. Bei höherer Reichweite wäre es möglich, dass mehrere Besucher parallel Informationen zu dem gleichen Objekt abrufen können.

Zudem kann es als störend empfunden werden, dass das Lesegerät nicht im PDA integriert ist. Durch zwei Geräte ist der Nutzer eingeschränkt, da diesem keine freie Hand mehr zur Verfügung steht. Wenn die Clientsoftware auf dem gleichen Gerät läuft, welches die IDs ausliest, erledigt sich das Problem. Bisher ist zwar noch kein PDA mit integriertem RFID-Lesegerät auf dem Markt zu finden, aber technisch ist dieses einfach zu realisieren. Alternativ wäre auch ein Lesegerät als SD-Karte nutzbar, so dass dieses als Erweiterungsmodul dem PDA verfügbar gemacht wird. Dabei müsste allerdings eine Anpassung im Sourcecode stattfinden, damit die richtige Schnittstelle angesprochen wird.

Merkliste

Eine andere sinnvolle Erweiterung ist eine Merkliste, mit der der Besucher die für ihn interessanten Objektinformationen verwalten kann. Sobald ein Objekt von Interesse ist, kann der Besucher dieses in einer Liste zwischenspeichern. Wenn er dann später zu diesem Objekt Informationen benötigt, kann er die ID aus der Liste entnehmen und Informationen vom Server abfragen.

Natürlich sollten nicht direkt die IDs in der Auswahlliste stehen. Stattdessen könnte man den Name oder auch gar die allgemeinen Informationen darstellen. Das System speichert intern zudem die ID, so dass Serverabfragen jeder Zeit möglich sind. Durch dieses Mapping ist dem Nutzer gleich klar, zu welchem Objekt der Eintrag gehört. Trotzdem geht keinerlei Funktionalität des Systems verloren.

Nutzungsstatistiken

Eine weitere mögliche Erweiterung des System ist eine Statistikfunktion im Server. Dabei kann neben der Erkennung von Abfragen mit unbekannter ID auch eine Zugriffsstatistik auf einzelne Datensätze erstellt werden. Hierdurch können die Mitarbeiter des Museum erkennen, welche Ausstellungsstücke bei den Besuchern besonders beliebt sind bzw. besonderes Interesse wecken.

Zudem wäre eine Erweiterung in diesem Bereich denkbar. Hierbei werden die Abfragen jedes einzelnen Clients gespeichert. Daraus kann dann entnommen werden, welche Ausstellungsstücke am häufigsten von gleichen Besuchern abgefragt werden. Mit diesen Informationen ist die Unterteilung einer Ausstellung an die Interessen der Besucher anpassbar. Zudem können bestimmte Interessensgebiete erkannt werden und z.B. auf diese mit entsprechenden Sonderausstellungen reagiert werden.

PC-Cafe

Als zusätzliche Erweiterung zur Merklis wäre es evtl. sinnvoll, das Museum um eine Art Cafe zu erweitern. Hier sind mehrere Computer zu finden, die den Besuchern zur Verfügung stehen. Dort soll eine entspannte Möglichkeit geboten werden, sich tiefergehend zu informieren.

Nach einem Besuch des Museums stehen verschiedene Objekte in der Merklis. Da sowohl das kleine Display des PDAs als auch die Tatsache, dass das Gerät getragen werden muss, den Komfort des Informierens einschränken, soll es möglich sein, die Merklis an einem Computer im Cafe zu übertragen. Danach kann der Gast den PDA abgeben und sich nach dem Besuch entspannen.

Zum Einen ist dadurch der PDA früher wieder verfügbar, so dass das Museum diesen schneller wieder einem anderen Besucher anbieten kann. Des Weiteren kann der Gast im Cafe verschiedene Informationen zu den Objekten seiner Merklis abrufen und so sich gezielt und bequem informieren.

Außerdem sind diese Rechner leistungsfähiger und bieten ein größeres Display. Dadurch sind auch multimediale Informationen möglich, die für einen PDA zu aufwändig wären. Somit wird dem Besucher eine individueller Möglichkeit der Informationsabfrage geboten. Das Lernen wird lockerer und dadurch auch für Menschen interessanter, die bisher eher selten in einem Museum anzutreffen sind.

Außerdem ist es möglich, dem Gast wie in einem normalem Cafe verschiedene Speisen anzubieten. Wenn die Computer entsprechend abgesichert sind, dass durch Flüssigkeiten oder andere Verunreinigungen der Hardware nicht geschehen kann, ist das System sicher und eine weitere Einnahmequelle für das Museum möglich.

Mobile Datenbank

Inzwischen gibt es in der Informatik Diskussionen und Projekte zu mobilen Datenbanken. Hierbei wird eine Datenbank nicht wie im klassischen Aufbau zentral durch einen Server gesteuert. Stattdessen wird die Datenbank auf die einzelnen Clients verteilt (siehe z.B. Thomé (2005)).

Durch eine solche Anpassung benötigen die Clients nicht durchgehend eine Verbindung zu einer Gegenstellen. Da Daten lokal verfügbar sind, können diese jeder Zeit abgerufen werden. Des Weiteren werden dadurch diese Abfragen wesentlich beschleunigt, was den Komfort verbessert. Allerdings wird für eine mobile Datenbank wesentlich mehr Speicher auf dem Client benötigt, was aber durch den Einsatz von SD-Speicherkarten erreicht werden kann.

J2ME-Client

Viele mobile Systeme bieten sehr begrenzte Ressourcen, weswegen dort häufig J2ME eingesetzt wird. Besonders bei aktuellen Handys ist diese Javaversion sehr häufig zu finden. Da schon Handys mit integriertem RFID-Lesegerät angeboten werden (siehe dazu Nokia (2004)), wäre es eine günstige Möglichkeit, dort die Clientsoftware einzusetzen.

Dem Besucher könnte die Möglichkeit angeboten werden, die Software auf sein eigenes Handy zu laden. Dadurch benötigt das Museum weniger Clients und der Besucher ist mit der Umgebung des Systems bereits vertraut. Allein die Serverabfrage müsste erweitert werden, so dass dieses per Email oder SMS möglich wird. Denkbar ist auch eine HTTP-Verbindung über Bluetooth, um direkt SOAP-Abfragen nutzen zu können. Dafür wird aber weitere Hardware benötigt, die die Verbindung zwischen dem J2ME-Client und dem Netzwerk regelt.

Somit müsste der Besucher nur die Software einspielen und ein Lesegerät erhalten. Schon ist ein weiterer Client einsatzbereit. Ein anderer Vorteil ist eine Merkliste über mehrere Museumsbesuche. Da der Gast die Software mitnehmen kann, kann er auch außerhalb des Museums seine Liste verwalten und dadurch seinen nächsten Museumsbesuch vorbereiten. Da diese Erweiterung naheliegend ist, wurde dieses auch im Sourcecode der entwickelten Clientsoftware berücksichtigt, so dass relativ geringe Anpassungen nötig sind.

Einkaufsbegleiter

Das System lässt sich aber auch in einem anderen Umfeld betreiben. Wenn Produkte eines Kaufhauses mit RFIDs ausgestattet sind, wäre es möglich einen Einkaufsbegleiter dem Kunden als Service anzubieten. Hierbei ist es möglich, verschiedene Informationen zu den einzelnen Produkten abzurufen. Somit könnte z.B. ein Kunde einfach und schnell alle Inhaltsstoffe zu einem Lebensmittelprodukt abfragen.

Zudem könnte sich jeder Kunde bei Interesse einen Account online erstellen, welcher dann durch den lokalen Client im Kaufhaus abgerufen wird. Dadurch ist z.B. eine Verwaltung einer Einkaufsliste möglich, welcher der Kunde im Vorfeld bequem von zu Hause aus verwalten kann. Dabei sind allerdings Datenschutz und Datensicherheit zu beachten, da nicht unbedingt personelle Daten von den Systemdaten getrennt werden.

Tiererkennung

Ein weiteres Einsatzgebiet wäre die Tiererkennung. Viele Haustiere bekommen inzwischen einen RFID-Chip implantiert, damit eine eindeutige Identifizierung des Tieres möglich ist. Dabei wird durch ein entsprechendes Gerät der Chip ausgelesen. Die dabei erkannte ID

muss allerdings zur Zeit telefonisch bei einer zentralen Datenverwaltungsstelle abgefragt werden.

Wenn diese Verwaltungsstelle ihr System um eine Serverabfrage erweitert, lässt sich MuseumAssistant an diese Umgebungsvorgaben anpassen. Dazu müssten die Serverabfragen angepasst werden und dem Client müsste ein Internetzugang zur Verfügung stehen. Letzteres lässt sich über ein Handy erreichen, welches per Bluetooth wie ein Modem angesteuert wird.

Eine komfortable Tiererkennung wäre dadurch möglich, die auch bei mobilen Einsätzen ihre Anwendung finden könnte. In Tierheimen würde aufgefundene Haustiere sofort erkannt werden und die Besitzer könnten innerhalb weniger Minuten benachrichtigt werden. Zudem wäre auch eine zentrale Verwaltungen der Krankenakten der Tier möglich, so dass im Notfall eine entsprechende Behandlung möglich ist.

Allerdings müsste für dieses System zuerst geklärt werden, wie weit das Datenschutzrechtlich realisierbar ist. Schließlich könnten dadurch die Akten der Tiere in einen direkten Zusammenhang zum Halter gebracht werden. Somit wäre evtl. nur die mobile Erkennung realisierbar, damit nicht gegen entsprechende Datenschutzgesetze verstoßen wird.

Glossar

A

ACL Asynchronous Connectionless, S. 16.

AFH Adaptive Frequency Hopping, S. 24.

B

Bluetooth Nahfunktechnik, S. 16.

C

CORBA Common Object Request Broker Architecture, S. 26.

D

DTD Document Type Definition, S. 41.

G

GAP Generic Access Profile, S. 21.

H

HCI Host Controller Interface, S. 20.

I

- IDL** Interface Definition Language, S. 27.
- IEEE** Institute of Electrical and Electronics Engineers, S. 23.
- ISM-Band** Industrial, Scientific and Medical band, S. 23.

J

- J2EE** Java 2 Platform, Enterprise Edition, S. 25.
- J2ME** Java 2 Platform, Micro Edition, S. 25.
- J2SE** Java 2 Platform, Standard Edition, S. 25.

K

- KVM** Kilobyte Virtual Maschine, S. 47.

L

- L2CAP** Logical Link Control and Adoption Layer Protokoll, S. 20.
- LMP** Link Manager Protokoll, S. 20.

M

- MIDP** Mobile Information Device Protocol, S. 25.

O

- OBEX** Object Exchange Protokoll, S. 21.
- ORB** Object Request Broker, S. 27.

P

PDA Personal Digital Assistant, S. 8.

R

RFID radio frequency identification, S. 9.

RMI Remote Method Invocation, S. 26.

S

SCO Synchronous Connection Oriented, S. 16.

SDAP Service Discovery Application Profile, S. 21.

SDP Service Discovery Protokoll, S. 20.

SIG Special Interest Group, S. 16.

SOAP Protokoll für Web-Services, S. 26.

SPP Serial Port Profile, S. 21.

SSID Service Set Identification, S. 24.

T

TCS Telephony Control Specification, S. 20.

U

UDDI Universal Description, Discovery and Integration, S. 28.

V

VoIP Voice over IP, S. 20.

W

WLAN Wireless Local Area Network, S. 10.

WSDL Web Service Description Language, S. 28.

Literaturverzeichnis

- Auf dem Hövel 2004** AUF DEM HÖVEL, Jörg: RFID: Hochkontrollierte Welten? In: *telepolis* (2004). – URL <http://www.heise.de/tp/r4/artikel/18/18845/1.html>. – Zugriffsdatum: 2005-04-06
- Böhmer u. a. 2005** BÖHMER, Reinhold ; BRÜCK, Mario ; REES, Jürgen: Die Welt in 96 Ziffern. In: *Wirtschaftswoche* (2005), Nr. 3, S. 36–47
- Bundesverband Informationswirtschaft 2004** BUNDESVERBAND INFORMATIONSWIRTSCHAFT, Telekommunikation und neue Medien e. V.: Informationspapier, Radio Frequency Identification (RFID) in der Diskussion. (2004). – URL http://www.bitkom.org/files/documents/BITKOM_RFID-Informationspapier_16.11.04.pdf. – Zugriffsdatum: 2005-01-18
- Burdet 2004** BURDET, Luc A.: RFID Multiple Access Methods. (2004). – URL http://www.vs.inf.ethz.ch/edu/SS2004/DS/reports/06_rfid-mac_report.pdf. – Zugriffsdatum: 2005-04-06
- Eberhart und Fischer 2001** EBERHART, Andreas ; FISCHER, Stefan: *Java-Bausteine für E-Commerce-Anwendungen*. 2., aktualisierte und erweiterte Auflage. Carl Hanser Verlag München Wien, 2001. – 191–252 S. – ISBN 3-446-21732-0
- Eberhart und Fischer 2003** EBERHART, Andreas ; FISCHER, Stefan: *Web Services*. 1. Auflage. Carl Hanser Verlag München Wien, 2003. – 175–238 S. – ISBN 3-446-22530-7
- eloqu 2005** ELOQU: mobilTour - Berliner Mauer. (2005). – URL http://www.eloqu.com/webpage_german/web_eloqu/projekte/mobil_tour.htm. – Zugriffsdatum: 2005-06-25
- Finkenzeller 2003** FINKENZELLER, Klaus: *RFID Handbook*. second edition. John Wiley & Sons, Ltd., 2003. – ISBN 0-470-84402-7
- Frank 2003** FRANK, Ulrich: Web Services und verteilte Datenbanken Simple Object Access Protocol. (2003)

- Jones 2004** JONES, Violet: Baja Beach Club in Barcelona, Spain Launches Microchip Implantation for VIP Members. (2004). – URL <http://www.prisonplanet.com/articles/april2004/040704bajabeachclub.htm>. – Zugriffsdatum: 2005-05-11
- Kauffels 2003** KAUFFELS, Dr. Franz-Joachim: *Lokale Netze Band 2*. 15.Auflage. mitp-Verlag/Bonn, 2003. – ISBN 3-8266-0994-8
- Kirst 200** KIRST, Martin: IRDA & Bluetooth. (200). – URL http://www-user.tu-chemnitz.de/~kirst/prosem/docs/b_prot2.htm. – Zugriffsdatum: 2005-06-09
- Löding 2005** LÖDING, Torge: RFID lässt Industrie träumen. In: *c't* (2005). – URL <http://www.heise.de/newsticker/meldung/56620>. – Zugriffsdatum: 2005-04-06
- Mahmoud 2002** MAHMOUD, Qusay H.: *Learding Wireless Java*. 1. Auflage. O'Reilly and Associates, Inc., 2002. – ISBN 0-596-00243-2
- Nokia 2004** NOKIA: Nokia RFID-Kit. (2004). – URL http://www.nokia.de/de/business/mobiles_geschaeftsleben/aussendienst/nokia_rfidkit/108266.html. – Zugriffsdatum: 2005-06-09
- OMG 2005** OMG: Getting Started with OMG specifications and Process. (2005). – URL <http://www.omg.org/gettingstarted/>. – Zugriffsdatum: 2005-04-20
- Postel u. a. 2004** POSTEL, Patrick ; SCHÜNEMANN, Sebastian ; ZDRZALEK, Jaroslaw: *Sicherheit in kommerziellen WLANSystemen*, HAW Hamburg, Diplomarbeit, 2004
- Schwarz 2004** SCHWARZ, Tobias: WLAN Standards und Spezifikationen. (2004). – URL <http://www.informationsarchiv.net/statisch/wlan/standards.html>. – Zugriffsdatum: 2005-02-20
- Stein 2005** STEIN, Martin: *RFID-Trackingsystem*, HAW Hamburg, Diplomarbeit, 2005
- SUN 2005** SUN: Java Remote Method Invocation (Java RMI). (2005). – URL <http://java.sun.com/products/jdk/rmi/>. – Zugriffsdatum: 2005-04-20
- Thomé 2005** THOMÉ, Mark: Mobile Datenbanksysteme. (2005). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2005/thome/abstract.pdf>. – Zugriffsdatum: 2005-06-18
- various 2003a** VARIOUS: Java 2 Platform, Micro Edition (J2ME). (2003). – URL <http://java.sun.com/j2me/index.jsp>. – Zugriffsdatum: 2005-03-18
- various 2003b** VARIOUS: PersonalJava. (2003). – URL <http://java.sun.com/products/personaljava/index.jsp>. – Zugriffsdatum: 2005-03-18

- various 2005a** VARIOUS: Anwendungsfälle und Szenarien für den Ferienklub. (2005).
– URL http://bart.informatik.haw-hamburg.de/wiki/index.php/Anwendungsfälle_und_Szenarien_für_den_Ferienklub. – Zugriffsdatum: 2005-06-25
- various 2005b** VARIOUS: Sun Java System RFID Software - Einstiegslösung SunStart. (2005). – URL <http://de.sun.com/servicesolutions/infrastructure/mobility/rfid/>. – Zugriffsdatum: 2005-06-25
- Wollert 2002** WOLLERT, Dr. Jörg F.: *Das Bluetooth Handbuch*. 1.Auflage. Franzis' Verlag, 2002. – ISBN 3-7723-5323-1
- Ziegler 2005** ZIEGLER, Peter-Michael: EU-Datenschutzgruppe führt Meinungsumfragen zu DRM und RFID durch. In: *c't* (2005). – URL <http://www.heise.de/newsticker/meldung/55921>. – Zugriffsdatum: 2005-04-06
- Zivadinovic 2004** ZIVADINOVIC, Dusan: Weltmeisterlich - Bluetooth-USB-Adapter mit großer Reichweite. In: *heise - mobil* (2004), 12. – URL <http://www.heise.de/mobil/artikel/54030/0>. – Zugriffsdatum: 2005-02-13
- Zivadinovic 2005** ZIVADINOVIC, Dusan: Enstörtes Verhältnis. In: *c't* (2005), Nr. 2, S. 126

A. DTD Kommandosatz

Die folgende DTD ist im Unterverzeichnis „readersets“ des Programmverzeichnisses in der Datei „readerset.dtd“ zu finden. Alle erstellten Kommandosätze müssen valide zu dieser DTD sein, damit das System damit arbeiten kann.

```
<!ELEMENT readerset (name, baudrate, databits, stopbits, parity, commandset)>
<!ELEMENT name (#PCDATA) >
<!ELEMENT baudrate (#PCDATA) >
<!ELEMENT databits EMPTY>
<!ATTLIST databits number (5 | 6 | 7 | 8) #REQUIRED>
<!ELEMENT stopbits EMPTY>
<!ATTLIST stopbits number (1 | 1_5 | 2) #REQUIRED>
<!ELEMENT parity EMPTY>
<!ATTLIST parity type (EVEN | MARK| NONE| ODD| SPACE) #REQUIRED>
<!ELEMENT commandset (initseq, commandseq+)>
<!ELEMENT initseq (commandseq)>
<!ELEMENT commandseq (name,
(commandtosend|commandtorec)*, (numberofreceivingbytes)?)>
<!ELEMENT commandtosend (parameter)+>
<!ELEMENT commandtorec (parameter)+>
<!ELEMENT numberofreceivingbytes (#PCDATA)>
<!ELEMENT parameter (#PCDATA)>
<!ATTLIST parameter type (byte | int | string | cwl) #REQUIRED>
```

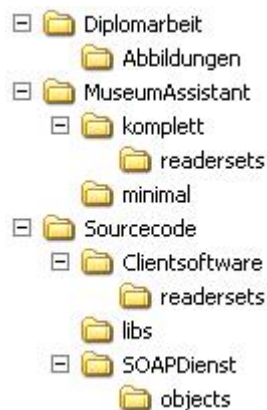
B. DTD Objectlist

Die folgende DTD ist im Unterverzeichnis „objects“ der Serversoftware in der Datei Objectlist.dtd zu finden. Alle Objektlisten müssen valide zu dieser DTD sein, damit die Objekte dem System bereitgestellt werden können.

```
<!ELEMENT ObjectList (Object)*>  
<!ELEMENT Object (ID, MainInfo, ExtendedInfo)>  
<!ELEMENT ID (#PCDATA)>  
<!ELEMENT MainInfo (#PCDATA)>  
<!ELEMENT ExtendedInfo (#PCDATA)>
```

C. CD-Inhalt

Folgende Abbildung zeigt die Verzeichnisstruktur der beigelegten CD-ROM.



Im Verzeichnis „Diplomarbeit“ finden Sie diese Arbeit in digitaler Form. Im Unterverzeichnis „Abbildungen“ sind alle Abbildungen dieser Arbeit zu finden.

Das Verzeichnis „MuseumAssistant“ beinhaltet zwei Versionen der Clientsoftware jeweils als jar-Datei. Die im Unterverzeichnis „komplett“ enthaltene Datei umfasst neben dem Bytecode auch benötigte jar-Archive sowie den Quellcode. Zudem ist im Verzeichnis „readersets“ die Datei mit dem Kommandosatz für den RFID-Leser der Firma SONMicro zu finden. Das Verzeichnis „minimal“ beinhaltet eine Clientsoftwareversion, die nur den benötigten Bytecode beinhaltet.

Im Verzeichnis „Sourcecode“ sind die einzelnen Quellcodes zu finden. Der SOAP-Dienst sowie die Clientsoftware sind jeweils im entsprechenden Verzeichnis zu finden. Im Unterverzeichnis „libs“ sind verschiedene Softwarebibliotheken zu finden, die für die Datenbank bzw. für den PDA benötigt werden.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(4) bzw. §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. Juni 2005

Ort, Datum

Unterschrift