



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht

Jan Napitupulu

Indoor Map Server in einem Flughafenszenario

Jan Napitupulu

Indoor Map Server in einem Flughafenszenario

Projektbericht im Rahmen der Veranstaltung Projekt

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Dipl. Inform. Birgit Wendholt

Abgegeben am 1. März 2007

Inhaltsverzeichnis

1	Einleitung	1
2	Konzeption	2
2.1	Systemarchitektur	2
2.2	Komponente IndoorLocationModell	2
2.2.1	Exemplarische Gebäudeinfrastruktur	4
2.3	Komponente LBSDirectory	4
2.3.1	Exemplarische Dienstinfrastruktur	5
2.4	Komponente RoutingComponent	6
2.5	Komponente IndoorMapModel	6
2.6	Schnittstellen	7
3	Realisierung	8
3.1	Verwendete Technologien	8
3.1.1	Entwicklungsumgebung	8
3.1.2	Zielplattformen	8
3.1.3	Positionsbestimmung mit IMAPS	9
3.1.4	EMIC Mapping and Location Framework	9
3.2	Probleme und Realisierungsentscheidungen	9
3.2.1	Einsatz des EMIC Location and Mapping Framework	10
3.2.2	Gewinnung der Positionsdaten vom IMAPS-Empfänger	10
3.2.3	Änderung der Zielplattform	10
3.2.4	Generierung der Gebäudekarte	11
3.3	Prototyp „PortGuide“	11
3.3.1	Systemanmeldung	11
3.3.2	Benutzertracking	12
3.3.3	Routenanforderung	12
3.3.4	Dienstanforderung	13
3.4	Weitere Arbeiten im Projekt	13
4	Fazit	14
A	Anhang: Schnittstellen	16

1 Einleitung

Der Aufenthaltsort eines Benutzers stellt bei ortsbezogenen Diensten (*location-based services*) eine besonders wichtige Information dar. Solche Dienste können beispielsweise Auskunft über bestimmte Dienstleistungen geben, die in der Umgebung des Anwenders zur Verfügung stehen (z.B. der nächstgelegene Geldautomat), oder aber auch selbst eine lokal verfügbare Dienstleistung bereitstellen. Die Metapher *Flughafen* bietet sich als exemplarisches Anwendungsfeld für solche Dienste besonders gut an: „Viele Menschen, die sich in einem großen Gebäudekomplex mit vielen unterschiedlichen *Points-of-Interest (POI)*, wie z.B. *Restaurants, Geschäfte oder Raucherbereiche, bewegen.*“

Mit diesem Hintergrund wurde im Rahmen des Projekts die Aufgabe verfolgt, ein System für die Anwendung in einem Flughafenszenario zu konzipieren und zu realisieren. Im Vordergrund standen dabei insbesondere die Positionsbestimmung innerhalb von Gebäuden, die Gewinnung von (ortsbezogenen) Auskünften zur Dienst- und Gebäudeinfrastruktur sowie die Personalisierung von Diensten. Die Vorarbeiten zu dem Projekt wurden bereits in [Koychev \(2006\)](#), [Kutak \(2006\)](#) und [Napitupulu \(2006\)](#) geleistet. Eine ausführliche Darstellung des Beispielsszenarios, das als Grundlage für die Arbeiten im Projekt gedient hat, findet sich in [Schumann \(2007\)](#).

Die Verantwortlichkeiten für die Teilsysteme des Projekts unterlagen dabei einer klaren Trennung. Die Abbildung 1 zeigt eine grobe Projektarchitektur, in dem die einzelnen Teilsysteme sowie deren Zusammenspiel skizziert werden.

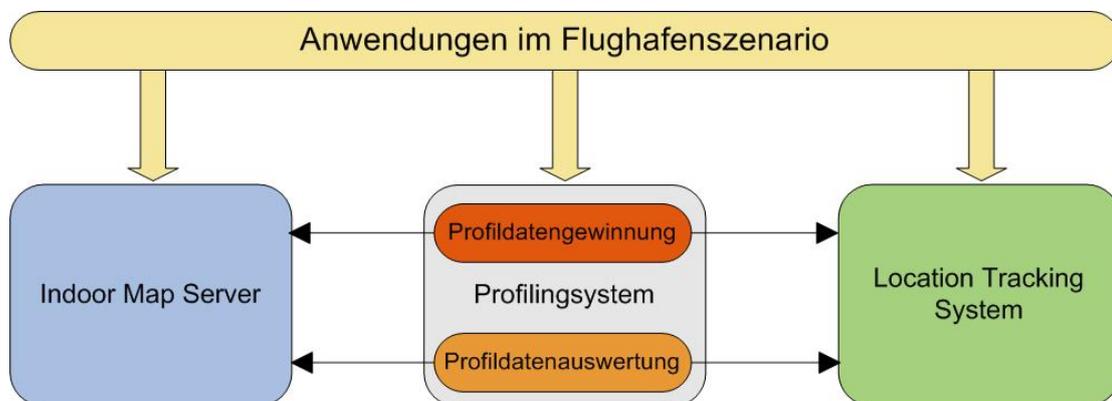


Abbildung 1: Einordnung in das Projekt

Die Hauptaufgabe des Autors im Rahmen des Projekts war die Konzeption und Realisierung eines *Indoor Map Server* ([Napitupulu, 2006](#)). Die vorliegende Ausarbeitung liefert einen zusammenfassenden Bericht zu den Arbeiten an diesem System, die im Verlauf des Projekts bewältigt wurden. In Kapitel 2 wird zunächst die Konzeption des *Indoor Map Server* beschrieben. Anschließend wird in Kapitel 3 auf die Realisierung des Systems eingegangen.

Neben den verwendeten Technologien werden dabei auch Probleme und daraus resultierende Realisierungsentscheidungen erläutert. Abgeschlossen wird dieses Kapitel durch die Vorstellung des Prototypen sowie einer kurzen Auflistung weiterer Arbeiten, die neben der eigentlichen Hauptaufgabe im Projekt erledigt wurden. In dem Fazit (Kapitel 4) werden die gewonnenen Erfahrungen zusammengefasst und ein Ausblick auf weitere Arbeiten in diesem Gebiet gegeben.

2 Konzeption

Eine grundlegende Analyse für die Anforderungen an den Indoor Map Server wurde bereits im Rahmen der Veranstaltung Anwendungen I durchgeführt (Napitupulu, 2006). Darauf aufbauend wird in diesem Kapitel die Konzeption vorgestellt, die im Projekt entwickelt und als Grundlage für die Realisierung (Kapitel 3) verwendet wurde.

2.1 Systemarchitektur

Wie in Napitupulu (2006) erläutert wurde, muss der hier geplante *Indoor Map Server* im Wesentlichen folgende Aufgaben erfüllen bzw. Möglichkeiten anbieten können:

1. Funktionalität eines (einfachen) Gebäudeinformationssystem
2. Bereitstellung eines Verzeichnis für ortsbezogene Dienste
3. Verknüpfung von Gebäudemodell und ortsbezogenen Diensten
4. Bereitstellung von Routeninformationen (z.B. für Navigationssysteme)

Mit diesen Vorgaben wurde ein System konzipiert, das die gewünschten Dienste zur Verfügung stellt. Ein erster Überblick der Architekturkomponenten ist der Abbildung 2 zu entnehmen. Die Funktionalität und die Aufgaben der einzelnen Komponenten werden in den folgenden Unterkapiteln kurz erörtert.

2.2 Komponente IndoorLocationModell

Eine wichtige Grundlage des *Indoor Map Server* ist der Zugriff auf die Gebäudeinformationen des Flughafens, wie z.B. das Gebäudemodell. Für die Verwaltung von solchen Informationen kommt in der Regel ein Gebäudeinformationssystem (GebIS) zum Einsatz. Dabei handelt es sich nach Jurgeit (2003) um ein System zur Erfassung, Dokumentation und Herausgabe von Gebäudeinformationen. Es umfasst ein Datenmodell, das sich in Geometrie- und Sachdaten unterteilen lässt. Bei den Geometriedaten handelt es sich zumeist um Vektor- oder Rasterdaten, mit denen die Gebäudebeschreibung in Drahtmodellen, Begrenzungsflächen oder

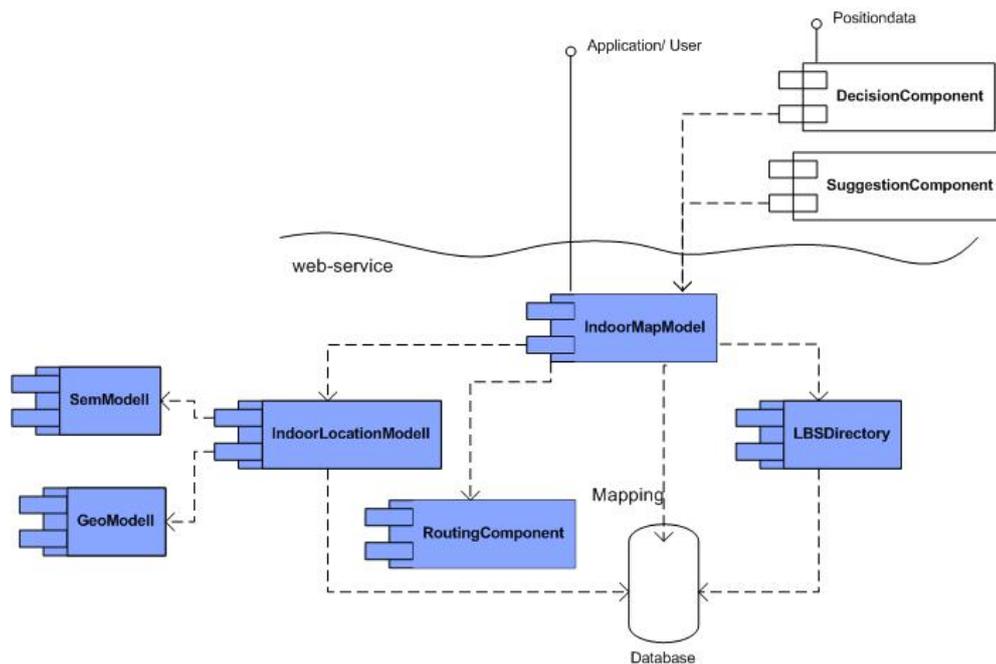


Abbildung 2: Architekturkomponenten des Indoor Map Server

Volumen vorgenommen wird (Zippelt, 1999). Die sachlichen oder auch semantischen Daten stellen eine (optionale) Erweiterung der Gebäudebeschreibung dar. Hiermit lassen sich beispielsweise Raum- oder Telefonnummern innerhalb eines Gebäudes hinterlegen.

Die Komponente *IndoorLocationModell* stellt in dieser Konzeption eine vereinfachte Form eines solchen Informationssystems dar. Die Gebäudedaten werden auch hier in geometrische und semantische Daten (*GeoModell* bzw. *SemModell*) unterteilt. Mit dem *GeoModell* werden die Räumlichkeiten der Gebäude als Polygone beschrieben. Bei den Polygonen wiederum handelt es sich effektiv um eine Liste von Punkten in einem Koordinatensystem. Für die Konzeption und Umsetzung im Rahmen des Projekts wurde hier zunächst nur Polygone bzw. Punkte in einem zweidimensionalen Koordinatensystem betrachtet. Über das *SemModell* wird einer geometrischen Fläche die entsprechende Bedeutung im Kontext eines Gebäudes zugewiesen, d.h. ob es sich beispielsweise um einen Raum oder um einen Durchgang handelt, der zwei Räume verbindet.

Für die Präsentation und Darstellung des Gebäudemodells in Anwendungen (z.B. einem Navigationssystem) kann mit dieser Komponente zudem eine zweidimensionale Grafik generiert werden, die einen Lageplan (in Draufsicht) darstellt. Bei der Anfrage eines solchen Lageplans werden zwei Informationen übermittelt: der eigentliche Lageplan als Grafik und eine XML-Datei mit Metainformationen zu dem Lageplan. Darin wird beispielsweise festgehalten, wie der Maßstab der Grafik ist (d.h. Meter/Pixel) oder wo die Bezugspunkte der Ecken liegen.

2.2.1 Exemplarische Gebäudeinfrastruktur

In Abbildung 3 wird der verwendete *virtuelle* Gebäudeplan dargestellt. Er besteht aus drei Räumen, einem Flur und drei Durchgängen bzw. Türen, die die Räume mit dem Flur verbinden.

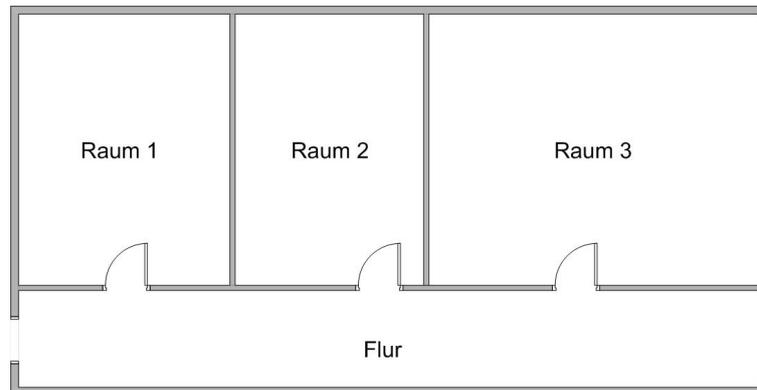


Abbildung 3: Lageplan der exemplarischen Gebäudeinfrastruktur

Die Öffnung zum Flur (Abbildung 3, unten links) wird hier nur zu illustrativen Zwecken verwendet, um den Eingang zum Flughafen zu kennzeichnen. Im Gebäudemodell wird diese jedoch semantisch nicht als Durchgang gewertet.

2.3 Komponente LBSDirectory

Die Komponente *LBSDirectory*¹ ist der Verzeichnisdienst für die ortsbezogenen Dienste in dem hier gewählten Szenario *Flughafen*. Schon in der frühen Phase der Konzeption wurde deutlich, dass bei dieser Komponente nicht die Frage „*Wie realisiert man einen Verzeichnisdienst?*“ im Vordergrund steht. Die Antwort auf diese Frage reduziert sich nämlich im Wesentlichen darauf, einfach eine Datenbank zu einzusetzen und die Anfragen an den Verzeichnisdienst in SQL-Queries umzuwandeln.

Viel interessanter bei diesem Teil des Systems ist die Beschreibung und Definition der Dienste, die letztendlich mit dem *LBSDirectory* verwaltet werden sollen. Im Rahmen des Projekts wurden folgende Informationen zur Beschreibung eines Dienstes verwendet:

ID Eine eindeutige Identifikationsnummer für den Dienst

Name Eine Benennung bzw. falls vorhanden der Eigenname des Dienstes (z.B. McDonald's)

TTU *Time-To-Use* ist die durchschnittliche Zeit, nach der man davon ausgehen kann, dass ein Benutzer im Bereich des Dienstes diesen auch genutzt hat.

¹LBS: location-based services

IMAPS im Labor so gewählt wurde, dass bei diesen beiden Räumen lediglich festgestellt werden kann, ob ein Benutzer den Raum betreten hat oder nicht. Der dritte Raum ist hauptsächlich als Gastronomiebereich vorgesehen. Im Gegensatz zu der Anordnung der Geschäfte auf dem Flur, soll dieser Raum repräsentativ für *weitläufige Bereiche* im Flughafen stehen, bei denen keine expliziten Wege vorhanden. Solche Bereiche stellen in der Praxis gerade bei Navigationssystemen für Fußgänger ein wesentliches Problem dar ([Tscheligi und Sefelin, 2006](#)) und wurden daher bei der Auswahl einer geeigneten Dienstinfrastruktur berücksichtigt.

2.4 Komponente RoutingComponent

Wie der Name schon erahnen lässt, werden über die Komponente *RoutingComponent* Möglichkeiten zur Routenberechnung für den Einsatz von Navigationssystemen im Flughafenszenario bereitgestellt. In einer ersten Konzeption wurde vorgesehen, dass der Routengraph basierend auf dem Gebäude- und Dienstmodell dynamisch berechnet werden sollte. Diese Thematik hat sich jedoch als zu komplex und zeitaufwendig erwiesen. Daher wurde entschieden, den Routengraph zunächst nur statisch zu hinterlegen.

Im Rahmen der Veranstaltung Anwendungen II wurde in [Napitupulu \(2007\)](#) unter anderem erörtert, dass insbesondere *Landmarken* das Mittel der Navigation für Fußgänger sind. Für den möglichen Einsatz eines Navigationssystems mit dem *Indoor Map Server* wurde daher entschieden, die Knoten für den Routengraphen wie folgt festzulegen:

1. Knoten auf Basis des Gebäudemodells:
Bei diesen Knoten handelt es sich um alle *Entscheidungspunkte* (z.B. Wegkreuzungen, Eingänge, Ausgänge) des Gebäudemodells. Bei dem hier gewählten Beispielszenario liegen die Entscheidungspunkte bei den Türen zu den Räumen 1-3.
2. Knoten auf Basis des Dienstmodells:
Alle verfügbaren Dienste können auch als *Landmarken* verwendet werden und werden daher als Knoten mit in dem Routengraph berücksichtigt.

Als Ergebnis einer Anfrage an diese Komponente kann damit eine Liste von Wegpunkten geliefert werden, die sich aus Entscheidungspunkten und Landmarken zusammensetzt. Auf Basis dieser Informationen ist es damit einem Navigationssystem, dass im Zusammenspiel mit dem *Indoor Map Server* betrieben wird, möglich, dem Benutzer sehr detaillierte und verständliche Navigationsinstruktionen zu liefern.

2.5 Komponente IndoorMapModel

In dieser Konzeption sorgt die Komponente *IndoorMapModel* zum einen dafür, dass die Dienste der Komponente *LBSDirectory* auf das Gebäudemodell der Komponente *Indoor-*

LocationModell abgebildet werden. Zum anderen steuert sie gesamten Ablauf und das Zusammenspiel der einzelnen Komponenten. Sämtliche Schnittstellen des *Indoor Map Server* laufen hier zusammen, so dass diese Komponente als exklusiver Zugriffspunkt auf das System fungiert.

Für die Abbildung der Dienste auf das Gebäudemodell werden die in Unterkapitel 2.3 dargestellten Dienstinformationen um ein Attribut erweitert:

Area Ähnlich wie bei der geometrischen Darstellung von Räumen handelt es sich hierbei um eine Liste von Punkten in einem Koordinatensystem, die insgesamt eine Fläche beschreiben können. Die Punkte bezeichnen hier absolute Positionen in dem Gebäudemodell.

Um feststellen zu können, ob sich ein Benutzer in Reichweite eines Dienstes befindet, werden zudem einige geometrische Berechnungsverfahren bereitgestellt, mit denen z.B. bestimmt werden kann, ob sich in Punkt (in einem zweidimensionalen Koordinatensystem) innerhalb einer Fläche (in einem zweidimensionalen Koordinatensystem) befindet.

2.6 Schnittstellen

Für die Kommunikation mit den anderen Teilsystemen und zur externen Nutzung des Systems durch weitere Anwendungen wurden einige Schnittstellen definiert. Die konkreten Schnittstellen wurden jeweils in verschiedenen Ausprägungen und unter Verwendung unterschiedlicher Parameter definiert, lassen sich aber grob in vier Kategorien einteilen:

1. Allgemeine Dienstanfragen:
Sämtliche verfügbare Dienste können direkt erfragt werden, z.B. unter Angabe der ID.
2. Positionsbezogene Dienstanfragen:
Hier können verfügbare Dienste beispielsweise unter Angabe einer Position oder eines Radius erfragt werden.
3. Visualisierung des Gebäudemodells:
In dieser Kategorie können zweidimensionale Lagepläne des Gebäudemodells angefordert werden, z.B. auf Basis der Position des Benutzers.
4. Routenanfragen:
Hierunter fallen alle klassischen Routenanfragen unter Angabe eines Starts und Ziels, evtl. noch mit Wegpunkten.

Sämtliche Schnittstellen laufen in der Komponente *IndoorMapModell* zusammen und werden über Webservices nach außen verfügbar gemacht. Eine Aufstellung der konkreten Schnittstellen zur Nutzung des Systems findet sich in Anhang A. Die Details zur technischen Umsetzung der Webservices werden in diesem Projektbericht nicht näher betrachtet und stattdessen auf [Koychev \(2007\)](#) verwiesen.

3 Realisierung

In diesem Kapitel werden die praktischen Arbeiten dargelegt, die im Verlauf des Projekts bewältigt wurden. Neben der Realisierung der in der Konzeption beschriebenen Hauptaufgabe, wird auch auf Probleme, Realisierungsentscheidungen und weitere Aufgaben im Projekt eingegangen.

3.1 Verwendete Technologien

3.1.1 Entwicklungsumgebung

Für die Realisierung des Projekts wurde gemeinsam frühzeitig entschieden, dass auf die Microsoft .NET Technologie gesetzt werden sollte. Dabei wurden das .NET Framework und das .NET Compact Framework in der Version 2.0 eingesetzt. Als Entwicklungsumgebung diente Microsoft Visual Studio 2005.

Für diese Technologieentscheidung sprachen mehrere Gründe. Beispielsweise sind alle benötigten Technologien über das Hochschulprogramm MSDNAA von Microsoft frei verfügbar. Da alle Produkte aus einem Haus kamen, wurde die Gefahr einer möglichen Inkompatibilität minimiert. Für .NET sprach aber auch die aktuelle Relevanz in der Praxis. Die meisten Projektteilnehmer haben bislang nur unter Java entwickelt. Das Projekt sollte daher dazu benutzt werden, zusätzliches Wissen und Qualifikationen für die Entwicklung unter .NET zu erlangen.

3.1.2 Zielplattformen

Alle Projektteilnehmer wurden mit einem Satz unterschiedlicher mobiler Geräte ausgestattet, die in Sachen Ausstattung und Kommunikationsmöglichkeiten dem aktuellen Stand der Technik entsprachen: ein Smartphone (*htc MTeoR pro*), ein PDA (*htc TyTN*) und ein Tablet PC (*Samsung Q1 Ultra Mobile PC*).

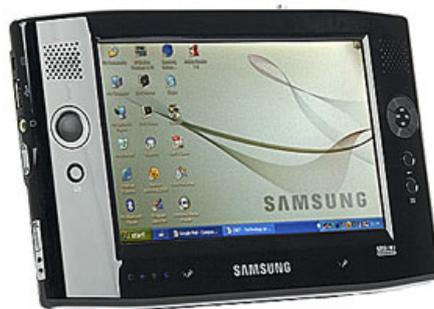


Abbildung 5: Zielplattform des Prototypen „PortGuide“

In der ursprünglichen Planung des Gesamtsystems wurde eigentlich der PDA als die gewünschte Zielplattform identifiziert. Solche Geräte genießen mittlerweile eine relativ hohe Verbreitung und sind gegenüber den meisten Smartphones zudem mit einer WLAN-Karte ausgestattet. Wie aber noch in Unterkapitel 3.2 beschrieben wird, gab es bei der Anbindung der Bibliothek für die Positionsbestimmung einige Probleme, so dass das hier vorgestellte Teilssystem zur Sicherheit für die Anwendung auf dem Tablet PC (Abbildung 5) entwickelt wurde.

3.1.3 Positionsbestimmung mit IMAPS

Für die Positionsbestimmung innerhalb von Gebäuden wurde IMAPS (Gregor, 2006) verwendet, das auch an der HAW Hamburg entwickelt wurde. Die Hauptarbeiten in diesem Gebiet formten ein eigenes Teilprojekt (Kutak, 2007).

Nichtsdestotrotz war es für die Nutzung von IMAPS in den Anfängen des Projekts als Gruppenleistung notwendig, die technische Infrastruktur zu schaffen, d.h. beispielsweise die Installation der IMAPS-Sender an der Decke des Labors. Die Details dieser Arbeiten sowie weitere anfallende Aufgaben in diesem Bereich werden in Kutak (2007) dargelegt.

3.1.4 EMIC Mapping and Location Framework

Neben der Realisierung des eigentlichen *Indoor Map Server* war eines der Ziele, eine geeignete Client-Anwendung umzusetzen, mit der die wesentlichen Funktionalitäten im praktischen Einsatz gezeigt werden können. Ein Teil dieser Anwendung sollte ein Tracking System sein, mit dem der Benutzer auf einer Karte des Gebäudes verfolgt werden. Darüber hinaus sollten etwaige verfügbare Dienste im Bereich des Benutzers visualisiert werden können.

Für diese Zwecke hat sich das *EMIC Location and Mapping Framework* von Microsoft (Microsoft, 2007) als sehr brauchbar erwiesen. Das Framework bietet viele Möglichkeiten zur Interaktion mit zweidimensionalen Karten, Darstellung von *Points-of-Interest* (bzw. *Areas-of-Interest*) und letztendlich zur Verfolgung des Benutzers auf einer Karte.

3.2 Probleme und Realisierungsentscheidungen

Bei der konkreten Realisierung des Systems musste an einigen Stellen von der ursprünglichen Konzeption (Kapitel 2) abgewichen und Änderungen vorgenommen werden. In diesem Unterkapitel werden diese Probleme und die daraus resultierenden Realisierungsentscheidungen erläutert.

3.2.1 Einsatz des EMIC Location and Mapping Framework

Grundsätzlich ist das Framework für die Nutzung in Outdoor-Szenarien unter Verwendung von GPS gedacht. Daher mussten vor dem eigentlichen Einsatz verschiedene Anpassungen vorgenommen werden. Eine wesentliche Änderung, die durchgeführt werden musste, war die Anpassung des so genannten *frame-of-reference*. Dabei handelt es sich effektiv um die Bezugseinheiten des zugrunde liegenden Koordinatensystems. Bei GPS werden Koordinaten üblicherweise in Längen- und Breitengraden angegeben. Während der gesamten Konzeption wurde jedoch stets davon ausgegangen, dass Positionen im Gebäudemodell in Metern zu einem fixen Bezugspunkt dargestellt werden. Daher war an dieser Stelle eine zusätzliche Umrechnung erforderlich. Zudem mussten die übermittelten Meta-Informationen zu einem Lageplan (vgl. Unterkapitel 2.2) geändert werden, da das EMIC Framework hier eine eigene Syntax erwartet.

3.2.2 Gewinnung der Positionsdaten vom IMAPS-Empfänger

Eines der größeren Probleme des Projekts war die tatsächliche Gewinnung der Positionsdaten vom IMAPS-Empfänger. Bei der Entwicklung des Positionierungssystem (Gregor, 2006) wurde eine Java-Bibliothek erstellt, mit der die entsprechenden Daten von der seriellen Schnittstelle empfangen und eine Positionsberechnung durchgeführt werden kann. Da im Projekt auf .NET gesetzt wurde, entstanden hier Probleme bei der Anbindung dieser Bibliothek. Nach einem gescheiterten Versuch die komplette Java-Bibliothek nach .NET zu konvertieren, wurde entschieden, die ursprüngliche Bibliothek einzusetzen und alle Positionsdaten über Interprozesskommunikation via TCP zwischen Java und .NET auszutauschen.

3.2.3 Änderung der Zielplattform

Wie schon zuvor erwähnt, war die gewünschte Zielplattform zunächst ein PDA. Der Prototyp für die Client-Anwendung wurde daher zunächst für dieses Gerät auf Basis des .NET Compact Framework realisiert. Mit der Problematik der Anbindung der Java-Bibliothek für den Zugriff auf den IMAPS-Empfänger musste hier jedoch umdisponiert werden. Zum einen war ungewiss, ob die Java-Bibliothek ohne weitere Probleme auf dem mobilen Gerät zum Laufen gebracht werden konnte. Zum anderen wurden die zwei parallel laufenden Java und .NET Anwendungen als zu ressourcenintensiv eingeschätzt. Daher wurde für den Prototyp dieses Teilprojekts der *UltraMobile PC* von Samsung als Zielplattform eingesetzt. Da dieser unter Windows XP betrieben wird, musste die Entwicklung mit dem .NET Framework (anstatt .NET Compact) neu aufgezogen werden.

3.2.4 Generierung der Gebäudekarte

In der ursprünglichen Konzeption war vorgesehen, dass die Gebäude- und Lagepläne, die vom *Indoor Map Server* angefordert werden können, serverseitig dynamisch generiert werden. Aufgrund der fortgeschrittenen Zeit und mangelnder Anwendungskennntnisse der dafür notwendigen .NET Bibliotheken, wurde dieses Feature jedoch als *nice-to-have* eingestuft und in dem vorliegenden System nicht berücksichtigt. Bei dem Aufruf der entsprechenden Schnittstelle liefert der *Indoor Map Server* daher momentan einen statisch hinterlegten Gebäudeplan.

3.3 Prototyp „PortGuide“

Zur Demonstration der Funktionalität des *Indoor Map Server* wurde eine prototypische Client-Anwendung auf Basis des .NET Frameworks entwickelt. Dieser Prototyp ist als „*Showcase*“ zur verstehen und nicht für die Benutzung durch den Endbenutzer gedacht.

3.3.1 Systemanmeldung

Zu Beginn erfolgt die Anmeldung des Benutzers an dem System (Abbildung 6). Der dafür notwendige Dienst wird über denselben Webservice angeboten, über den auch die Schnittstellen des *Indoor Map Server* zur Verfügung stehen. Die Systemanmeldung ist für die Nutzung des *Indoor Map Server* eigentlich nicht erforderlich, wird aber für das Profilingssystem benötigt, dass von [Schumann \(2007\)](#) und [Koychev \(2007\)](#) behandelt wurde.

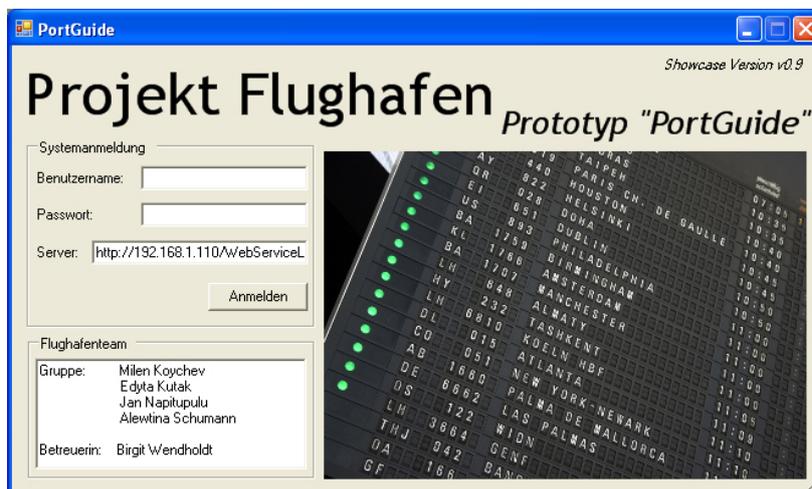


Abbildung 6: Prototyp „PortGuide“: Systemanmeldung

3.3.2 Benutzertracking

Eine Funktion des Prototypen ist das Benutzertracking. Dazu wird der Gebäudeplan vom Server angefordert und mit Hilfe des *EMIC Location and Mapping Framework* angezeigt (Abbildung 7). Sofern der IMAPS-Empfänger angeschlossen ist, wird der Benutzer entsprechend seiner Position auf der Karte dargestellt. Zu Testzwecken kann das System aber auch ohne reale Positionsdaten betrieben werden. Der Benutzer kann dann *per Mausclick* auf der Karte bewegt werden.

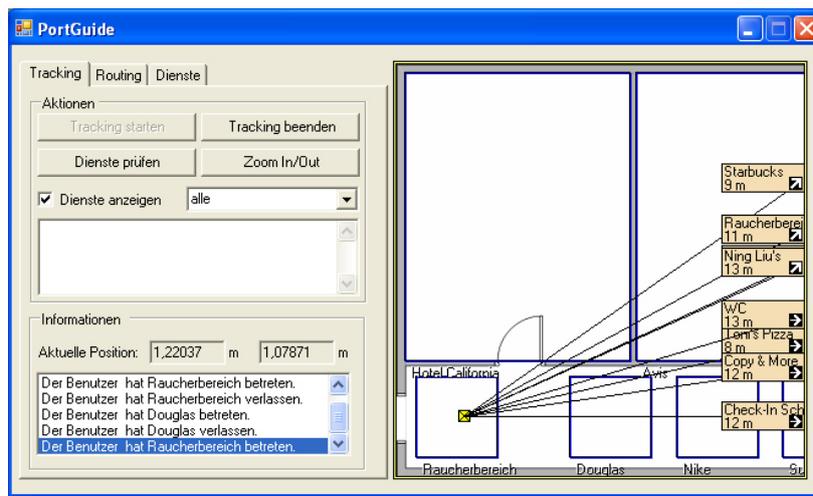


Abbildung 7: Prototyp „PortGuide“: Tracking des Benutzers (mit Ansicht aller Dienste)

Optional lassen sich noch alle verfügbaren Dienste einblenden. Dienste, die außerhalb des Darstellungsbereich liegen, werden durch Wegweiser gekennzeichnet, die Auskunft über Richtung und Entfernung geben.

3.3.3 Routenanforderung

In der ersten Arbeit zu diesem Thema (Napitupulu, 2006) war geplant, eine Indoor-Navigationssystem zu realisieren. Dieses Ziel konnte als solches aus zeitlichen Gründen nicht erreicht werden. Dennoch wurden durch die Komponente *RoutingComponent* (Unterkapitel 2.4) schon einige wesentliche Grundsteine dafür gelegt. Um dies zu zeigen, lässt sich mit dem Prototypen (unter Angabe eines Start- und eines Zielpunktes) ein Route anfordern und auf dem Gebäudeplan darstellen (Abbildung 8).

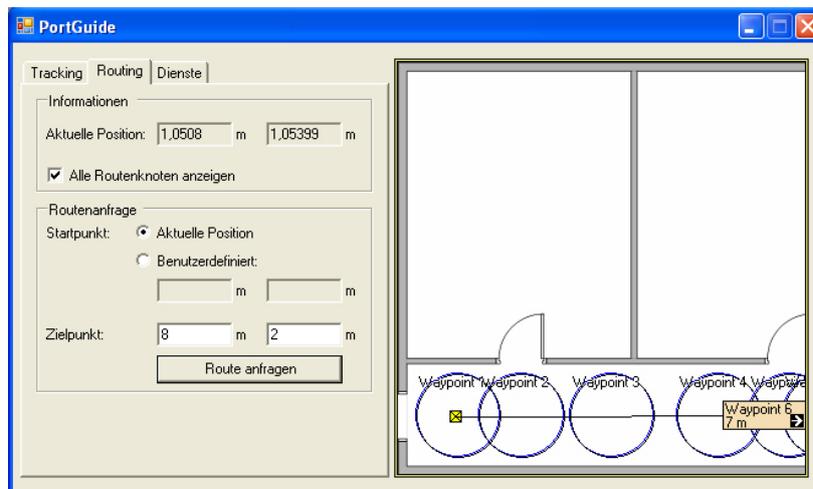


Abbildung 8: Prototyp „PortGuide“: Routenanforderung

3.3.4 Dienstanforderung

Die Anforderung von Informationen zu verfügbaren Diensten ist eine der Hauptaufgaben des Indoor Map Server und wird daher ebenfalls durch den Prototypen veranschaulicht. In der vorliegenden Version sind drei unterschiedliche Dienstanforderungen möglich:

- alle verfügbaren Dienste anfordern
- Dienste an einer bestimmten Position anfordern
- Dienste im Umkreis einer bestimmten Position anfordern

In Abbildung 9 wurden beispielsweise alle Dienste in einem Radius von zwei Metern um die aktuelle Benutzerposition angefordert.

3.4 Weitere Arbeiten im Projekt

Neben der Konzeption und Realisierung des Indoor Map Server sowie des zuvor vorgestellten Prototypen der Client-Anwendung sind im Projekt weitere Arbeiten angefallen, die ebenfalls Zeit in Anspruch genommen haben und daher auch erwähnenswert sind. Diese Arbeiten wurden teilweise alleine und teilweise mit einem bzw. mehreren Projektteilnehmern bewältigt, was im Folgenden durch die Begriffe *Einzelarbeit* und *Gruppenarbeit* gekennzeichnet wird:

- *Gruppenarbeit*: Laboraufbau, d.h. Aufbau der Infrastruktur für IMAPS, Einrichtung der Arbeitsplatzrechner (Betriebssysteme, Entwicklungsumgebungen,...), Netzwerkkonfiguration

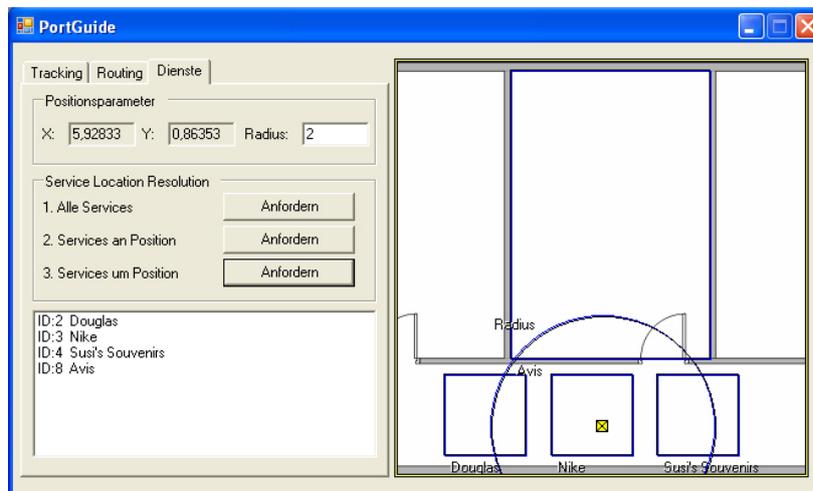


Abbildung 9: Prototyp „PortGuide“: Dienstanforderung nach Positionsauflösung

- *Gruppenarbeit*: Genaues Vermessen des Labors und Erstellung des exemplarischen Gebäudeplans
- *Einzelarbeit*: Installation und Konfiguration eines Wikis für das Projekt (dessen Nutzung sich im Verlauf leider totgelaufen hat)
- *Einzelarbeit*: Installation und Konfiguration eines Systems zur Versionskontrolle (zuerst CVS, später Umsatlung auf SVN)
- *Gruppenarbeit*: Anpassung der Positionsberechnungsalgorithmen der Java-Bibliothek von IMAPS für die Verwendung in der konkreten Infrastruktur der IMAPS-Sender im Labor
- *Gruppenarbeit*: Optimierung der Interprozesskommunikation zwischen der Java-Bibliothek und dem Prototypen

Insbesondere der Laboraufbau hat in der Anfangsphase des Projekts einiges an Zeit gekostet, so dass die tatsächliche *Produktivitätsphase* erst sehr viel später als geplant beginnen konnte.

4 Fazit

Durch den Projektverlauf und die erzielten Ergebnisse konnten zusammenfassend folgende Erfahrungen gesammelt werden:

- Die Positionsbestimmung mit IMAPS ist noch verbesserungswürdig. Es wurde deutlich, dass das System zu selten brauchbare Positionen abliefert. Bei der Tracking Funktion des Prototypen äußert sich das dadurch, dass die Aktualisierung der Benutzerposition stets zeitlich verzögert geschieht und damit kein echtes *Realtime-Tracking* möglich ist.
- Die (frühzeitige) Festlegung auf die .NET Technologie brachte einige Probleme mit sich. So hat die Einarbeitungsphase wesentlich länger gedauert, als ursprünglich vorgesehen war. Hier wäre eine kritischere Betrachtung der zur Verfügung stehenden Zeit vonnöten gewesen. Hätte man zudem von Anfang an auf Java gesetzt, wäre auch die Anbindung der Bibliothek zur Ansteuerung der IMAPS-Empfänger nicht zu so einem großen Problem gewachsen, wie es in der Endphase des Projekts der Fall war.
- Sämtliche Vorarbeiten für die eigentliche Hauptaufgabe im Projekt haben sehr viel Zeit in Anspruch genommen.
- Eine Beschreibung bzw. Definition der zur Verfügung stehenden Dienste ist nicht trivial. Inwieweit die im Rahmen des Projekts gewählten Informationen zur Beschreibung eines Dienstes (Unterkapitel 2.4) im praktischen Einsatz tatsächlich ausreichend sind, konnte nicht festgestellt werden.

In Bezug auf die ursprüngliche Analyse ([Napitupulu, 2006](#)) und auf die Zielsetzungen zu Beginn des Projekts kann insgesamt festgehalten, dass ein zufrieden stellendes Ergebnis erzielt wurde. Speziell in Bezug auf den Indoor Map Server konnte allen wesentlichen Anforderungen an die Gebäude- und Dienstverwaltung zumindest in rudimentärer Form genüge getan werden. Nichtsdestotrotz bleibt natürlich noch viel Raum für Verbesserungen und weitere Experimente. So ist das hier konzipierte einfache Gebäudeinformationssystem und die Visualisierung mit dem *EMIC Location and Mapping Framework* nur eine moderate Lösung. Für den praktischen Einsatz in einem realen Umfeld sollte man hier auf professionellere Lösungen zurückgreifen. Ein Aspirant dafür ist beispielsweise [CityGML \(2007\)](#), ein semantisches Datenformat für Stadt- und Gebäudemodelle. Neben der reinen Datenhaltung gibt es für diesen Standard auch einige Java-basierte Tools, die eine zwei- und dreidimensionale Visualisierung ermöglichen, wie z.B. [Aristoteles \(2007\)](#).

Ein Hauptaugenmerk des gesamten Projekts galt der Untersuchung inwieweit *IMAPS* mit seinem momentanen Entwicklungsstand für die Praxis geeignet ist. Wie bereits erwähnt, wurde diesbezüglich festgestellt, dass die Positionsbestimmung zwar grundsätzlich funktioniert, aber noch stark verbesserungswürdig ist. Daher ist es hier empfehlenswert, zunächst die Genauigkeit und Zuverlässigkeit von *IMAPS* weiterzuentwickeln, bevor weitere Projekte gestartet werden, die auf diesem System aufbauen.

A Anhang: Schnittstellen

- **GetMap2DbyName**
Liefert die zweidimensionalen Daten des Gebäudes, über Name und Ebene
- **GetMap2DbyPos**
Liefert die zweidimensionalen Daten des Gebäudes, über eine (Benutzer-)Position
- **GetNextService**
Liefert den am nächsten liegenden Dienst
- **GetNextServices**
Liefert eine Liste mit den am nächsten liegenden Diensten
- **GetNextServicesCategory**
Liefert eine Liste mit den am nächsten liegenden Diensten einer bestimmten Kategorie
- **GetRouteByPos**
Liefert eine Route (in Form von Routeknoten) zwischen zwei Positionen
- **GetServiceByID**
Liefert den Service mit der gegebenen ID
- **GetServices**
Liefert alle eingetragenen Services
- **GetServicesAround**
Liefert alle Services innerhalb des Umkreises einer gegebenen (Benutzer-)Position
- **GetServicesAt**
Liefert alle Services für eine gegebene (Benutzer-)Position
- **GetServicesByID**
Liefert alle eingetragenen Services mit den gegebenen IDs

Literatur

- [Aristoteles 2007] ARISTOTELES: *Aristoteles GML3D Viewer Official Website*. 2007. – URL <http://131.220.71.208/index.php/Aristoteles>. – Zugriffsdatum: 10.02.2007
- [CityGML 2007] CITYGML: *CityGML Official Website*. 2007. – URL <http://www.citygml.org/>. – Zugriffsdatum: 10.02.2007
- [Gregor 2006] GREGOR, Sebastian: *Entwicklung einer Hardwareplattform für die Ermittlung von Positionsdaten innerhalb von Gebäuden*. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gregor.pdf>. – Zugriffsdatum: 20.02.2007
- [Jurgeit 2003] JURGEIT, Florian: *Rauminformationssystem mit PostgreSQL und SVG*, Universität Innsbruck, Diplomarbeit, 2003. – URL http://www.carto.net/papers/florian_jurgeit. – Zugriffsdatum: 27.02.2007
- [Koychev 2006] KOYCHEV, Milen: *Personalisieren von Diensten*. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/koychev/abstract.pdf>. – Zugriffsdatum: 28.02.2007
- [Koychev 2007] KOYCHEV, Milen: *Master Projekt Flughafen*. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-proj/berichte.html>. – Zugriffsdatum: 28.02.2007
- [Kutak 2006] KUTAK, Edyta: *Entwicklung eines Location Tracking Systems für die Indoor Navigation*. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/kutak/abstract.pdf>. – Zugriffsdatum: 28.02.2007
- [Kutak 2007] KUTAK, Edyta: *Hardwareplattform für die Positionsbestimmung innerhalb von Gebäuden*. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-proj/berichte.html>. – Zugriffsdatum: 28.02.2007
- [Microsoft 2007] MICROSOFT: *EMIC Location and Mapping Framework*. 2007. – URL <http://www.microsoft.com/emea/emic/downloadcentre.msp>. – Zugriffsdatum: 14.02.2007
- [Napitupulu 2006] NAPITUPULU, Jan: *Indoor Map Server in einem Flughafenszenario*. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/napitupulu/abstract.pdf>. – Zugriffsdatum: 20.02.2007

- [Napitupulu 2007] NAPITUPULU, Jan: *Multimediale Fluggastführung*. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/napitupulu/report.pdf>. – Zugriffsdatum: 27.02.2007
- [Schumann 2007] SCHUMANN, Alewtina: *Indoornavigationssystem in Flughafen-methapher*. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-proj/berichte.html>. – Zugriffsdatum: 28.02.2007
- [Tscheligi und Sefelin 2006] TSCHELIGI, Manfred ; SEFELIN, Reinhard: Mobile navigation support for pedestrians: can it work and does it pay off? In: *interactions* 13 (2006), Nr. 4, S. 31–33. – URL <http://doi.acm.org/10.1145/1142169.1142192>. – Zugriffsdatum: 09.02.2007. – ISSN 1072-5520
- [Zippelt 1999] ZIPPELT, Karl: Aspekte der Datenmodellierung in Gebäudeinformationssystemen. In: *DVW-Seminar Gebäudeinformationssysteme*, URL <http://www1.gik.uni-karlsruhe.de/~zippelt/pub/gebis.html>. – Zugriffsdatum: 06.06.2006, 1999