

# Carpe Guide - ein barrierefreier, kontextsensitiver Multimedia-Guide für mobile Geräte

## Carpe Guide – a barrier-free, context-sensitive multimedia guide for mobile devices

Eileen Kühn<sup>1</sup>, Ronny Pflug<sup>1</sup>, Maurus Rohrer<sup>1</sup>, Elisabeth Sieck<sup>2</sup>, Jürgen Sieck<sup>1</sup>

<sup>1</sup> Hochschule für Technik und Wirtschaft Berlin  
Treskowallee 8, 10318 Berlin

<sup>2</sup> Humboldt-Universität zu Berlin  
Unter den Linden 6, 10099 Berlin

E-Mail: eileen.kuehn@htw-berlin.de, ronny.pflug@student.htw-berlin.de, mrohrer@hsr.ch,  
E.Sieck@gmx.de, j.sieck@htw-berlin.de

### **Zusammenfassung:**

Im Beitrag werden der Entwurf und die Implementierung eines barrierefreien, kontextsensitiven Multimedia-Guides für mobile Geräte präsentiert. Durch die Zusammenarbeit von Studierenden der Hochschule für Technik und Wirtschaft Berlin und Humboldt Universität Berlin mit Mitarbeitern von Carpe Berlin sind neue Ideen entwickelt worden, wie durch die Nutzung von Kontextinformationen und neuer Nutzerschnittstellen barrierefreie Multimedia-Guides für mobile Geräte realisiert werden können. Ziel der Entwicklung war es, blinden und sehbehinderten Menschen einen Zugang zu multimedialen Informationen in Stadt- und Tourismusinformationssystemen mit Hilfe von mobilen Geräten zu ermöglichen.

### **Abstract:**

The paper presents the design and implementation of a barrier-free, context-sensitive multimedia guide for mobile devices. In a cooperation between students of the University of Applied Sciences Berlin (HTW Berlin) and Humboldt University Berlin and employees of Carpe Berlin new ideas on how to create barrier-free multimedia guides for mobile devices with the usage of context information and new user interfaces were developed. The intention of the development was to provide blind and visually impaired people access to multimedia information from city and tourist information systems with mobile devices.

### **Einleitung**

Befindet man sich als Besucher in einer neuen Umgebung so gibt es oft keine Möglichkeit selbständig an Information zum gerade Gesehenen zu gelangen. Besonders in einer Metropole wie Berlin entgehen einem so viele interessante Fakten und Informationen. Häufig werden Audio-Guides angeboten, deren Bedienung oft sehr kompliziert ist. Viel geeigneter wäre ein Touristenführer dem man nur noch interessiert lauschen muss, ohne die gewünschten Informationen selbst zu suchen und das mit einem Gerät, das intuitiv und barrierefrei (auch für Blinde und Sehbehinderte) bedienbar ist.

Da normale Audio-Guides für die Masse jedoch denkbar unpraktikabel sind, wurde ein Konzept für einen Multimedia-Guide entwickelt, der den Nutzer nahezu vollautomatisch mit Information in Abhängigkeit seiner aktuellen Position und Situation versorgt. Ausgehend von dieser Idee wurde für ein Mobiltelefon eine kontext- und positionssensitive Multimedia-Anwendung entwickelt. Die besondere Herausforderung bestand in der barrierefreien, zumindest aber barrierearmen Umsetzung. Darüber hinaus sollten auch Menschen mit einer Sehbehinderung oder einer vollständigen Erblindung dieses System nutzen können.

Ausgangspunkt der Überlegungen war Paragraph § 4 des „Gesetzes zur Gleichstellung behinderter Menschen“. In ihm heißt es:

*„Barrierefrei sind [...] Systeme der Informationsverarbeitung [...], wenn sie für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.“*

Die zentrale Frage der Gesamtentwicklung war, ob die Barrierefreiheit durch die Minimierung von Nutzereingaben, die Nutzung der in Mobiltelefonen verbauten Sensoren und neuartige Bedienkonzepte zu gewährleisten ist.

### Ziele und Funktionalität des Systems

Ziel des Systems ist es, dem Nutzer die Möglichkeit zu geben sich während seiner Reise über seine Umgebung informieren zu lassen. Ist der Carpe Guide erst ein mal gestartet beginnt sofort die Positionsbestimmung. Dies geschieht in sehr kurzen Abständen. So weiß der mobile Client ob und in welche Richtung er sich bewegt. War die Positionsbestimmung erfolgreich, kontaktiert der Client den Server und erhält alle verfügbaren Medien auf Basis seiner momentan Position. Befindet sich der Client nun an einer Position, an der multimediale Informationen hinterlegt worden ist, spielt er dieses automatisch ab. Über welche Themen der Nutzer informiert wird, kann er selbst bestimmen. Er kann sich für ein oder mehrere Kategorien interessieren und bekommt so den speziell auf ihn zugeschnittenen Inhalt geliefert. Zudem hat der Benutzer die Möglichkeit die Inhalte selbst zu steuern. Er kann Vor- und Zurückspulen und hat die Möglichkeit das Abspielen zu unterbrechen, es wieder zu starten oder aber abzubrechen. Ermittelt der Client verfügbare Informationen bekommt der Nutzer ein optisches Feedback, um welchen Ort es sich handelt. Befindet er sich gar auf einer vorher definierten Route, so bekommt er eine Information über den letzten und den darauffolgenden Ort und auf welcher Route er sich befindet.

Das Pflegen der Inhalte wird über ein Autorensystem, ein s.g. Content Management System, bewerkstelligt. Hier hat der Autor die Möglichkeit neue Orte einzupflegen, sie zu editieren und sie wieder zu löschen. Jedem Ort können dann mehrere unterschiedliche Informationen hinterlegt werden. Jeder Einzelinformation kann außer dem auch noch eine Route zugeordnet werden.

### Technische Konzepte und Features des Clients

Um eine effektive Implementierung der erforderlichen Funktionalitäten zu gewährleisten, wurde der Carpe Guide als eine Client-Server-Architektur umgesetzt. Als Client kommen das iPhone von Apple zum Einsatz, da es die für die Anwendung nötige Sensorik zur Verfügung stellt. Realisiert wurde die Anwendung als Model-View-Controller (MVC) Architektur, welche sich an die von der Firma Apple modifizierte Version der MVC Architektur anlehnt (siehe Abb. 1). Da das iPhone mit einer speziellen Version des Betriebssystems Mac OS X läuft, wurde der Multimedia Guide mit der Programmiersprache Objective-C 2.0 und dem iPhone SDK 2.2 umgesetzt.

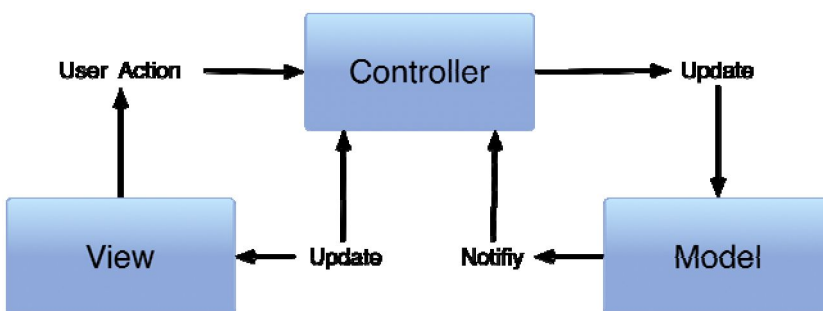


Abbildung 1: MVC-Prinzip lt. Apple Inc.

Intensive Überlegungen und Untersuchungen galten der Gestaltung der Bedienungs Oberfläche, von der mehrere Versionen entwickelt und evaluiert wurden. Einige hatten einen sehr hohen gestalterischen Anspruch, vernachlässigten jedoch die intuitive Bedienung und insbesondere die Benutzerfreundlichkeit für behinderte Menschen war

nicht zu gewährleisten. So war z.B. die Integration von Bildern und Texterläuterungen, mit welcher S-Bahn man gerade fährt und welche Möglichkeiten zur Weiterreise existieren sehr problematisch, da gerade sehbehinderten oder blinden Nutzern diese visuellen Informationen nicht zugänglich

sind. Auch die Farbwahl und ein geeignetes Oberflächendesign musste immer wieder überarbeitet werden, da Sehbehinderungen wie beispielsweise Rot-Grün-Schwäche, das Bild so verändert haben, dass das ursprüngliche Layout nicht mehr zu erkennen war. Im Ergebnis dieser Untersuchungen haben wir uns für ein sehr einfacheres und intuitiv zu bedienendes User Interface entschieden (siehe Abb. 2). Nach dem Testen mehrere Farbkombinationen wurde als Basisfarbe Blau gewählt, da hier die gesamte Oberfläche die wenigsten Abweichungen zum tatsächlichen Carpe Guide Layout zeigte.

Um das barrierearme Konzept des User Interfaces auch den Nutzern mit Sehbehinderung zugänglich zu machen, wurden die im iPhone verfügbaren Sensoren genutzt und auf diese Weise eine alternative Steuerung geschaffen. Diese basiert hauptsächlich auf dem Prinzip einer Gestensteuerung. Um dies zu realisieren, machen wir Gebrauch von den Beschleunigungs- und Berührungssensoren des iPhones. Der Berührungssensor kommt bei der Steuerung des Audio-Players zum Einsatz. Hierbei wurde bei der Umsetzung explizit auf die bereits von Apple etablierten Gesten Rücksicht genommen, um die Usability der Applikation für den Nutzer zu maximieren. Zum Vor- und Zurückspulen haben wir die sogenannte Swipe-Geste speziell an unsere Anforderungen angepasst und implementiert. Wischt der Nutzer mit einem Finger auf dem Bildschirm von links nach rechts, wird der multimediale Inhalt vorwärts gespult und wischt man von rechts nach links, wird zurückgespult. Der Inhalt wird mit dem Double-Tap, einem doppelten Tippen mit dem Finger, pausiert bzw. wieder gestartet.

Um die komplette Bildschirmfläche des iPhones (320x480px) optimal auszunutzen, können die Gesten auf der kompletten Fläche des Touchscreens ausgeführt und ausgewertet werden. Dies bietet speziell im Hinblick auf die barrierearme Umsetzung einen großen Vorteil: So kann jede Geste überall auf dem Bildschirm ausgeführt werden, ohne ein explizit für die jeweilige Funktion vorgesehenes Steuerelement berühren zu müssen.

Um auch das Abbrechen von Audioinformationen oder anderen Aktionen an jeder beliebigen Position im Programm zu ermöglichen, benutzen wir zusätzlich zu den Berührungssensoren den 3D-Beschleunigungssensor des iPhones. Dieser erlaubt es, die Bewegung des iPhones genaustens zu ermitteln und zu analysieren. Die Auswertung der Beschleunigungssensordaten erlaubt die Implementierung einer Schüttel-Geste, die zur Steuerung des Systems genutzt werden kann. Wird das iPhone zum Beispiel mehrmals geschüttelt, bricht der Carpe Guide die momentan ausgeführte Aktion ab. Dies gilt sowohl in gesonderten Menüs innerhalb der Applikation als auch für das Abspielen von Audioinhalten. Damit das normale Umherlaufen des Benutzers nicht fälschlicherweise als Schüttel-Geste wahrgenommen und interpretiert wird, mussten unterschiedliche Bewegungsmuster in der S-Bahn und zu Fuß analysiert werden. Als Ergebnis dieser Untersuchungen wurde ein Beschleunigungswert von  $1.2\text{ g}$  ( $1\text{g} \approx 9.81\text{ m/s}^2$ ) ermittelt, der als Schwellwert zur Erkennung des Schüttelns genutzt wird.

Um das ganze Konzept abzurunden und den Nutzer optimal auf die Bedienung des Guides einzustimmen, führt eine speziell entwickelte Audio-Sequenz den Nutzer beim Start der Applikation durch die grundlegende Bedienung des Carpe Guides. Die Audio-Einführung zeigt die Möglichkeiten der Gestensteuerung auf und bezieht den Benutzer sofort aktiv mit ein. Es wird jede Geste separat erläutert und anschließend auf die erfolgreiche Ausführung des Nutzers gewartet. Als Feedback der Eingabe ertönt eine Erfolgs- oder eine Fehlermeldung. So hat der Benutzer noch vor dem eigentlichen Start des Carpe Guides die Möglichkeit die komplette Gestik kennen zu lernen und auszuprobieren.



Abbildung 2: Client-Ansicht

Als letzte Unterstützung dient ein haptisches Feedback, in Form einer Vibration des iPhones, um anzuzeigen dass eine nicht verfügbare bzw. nicht definierte Geste ausgeführt worden ist.

Die multimedialen Inhalte, die dem Nutzer während seiner Tour durch die Stadt zur Verfügung gestellt werden, werden positions- und situationsbezogen von einer zentralen Instanz, dem Server, heruntergeladen. Auf diese Weise können zudem stets aktuelle Daten und Informationen zur Umgebung des Nutzers bereitgestellt werden. Um neben der Aktualität der Inhalte auch die Kosten für den Benutzer zu minimieren, werden die heruntergeladenen Daten auch über mehrere Applikationsstarts hinaus zwischengespeichert und können so bei Bedarf erneut angehört werden.

### Umsetzung des Servers

Der Server hat unterschiedliche Dienste bereitzustellen. Er ist beispielsweise für die zentrale Datenhaltung der Inhalte verantwortlich. Alle für das System notwendigen Daten werden dazu konsistent in einer MySQL Datenbank gespeichert. Zusätzlich stellt der Server ein Content Management System zur Verfügung, mit dessen Hilfe die Inhalte durch autorisierte Personen erfasst, ergänzt, geändert und gelöscht werden können.

Eine sehr wichtige Komponente des Servers ist die Kommunikations-Schnittstelle zum Client. Auf Grundlage dessen kann der Carpe Guide die benötigten Daten ermitteln und diese abrufen. Um dies zu realisieren, übermittelt der Client dem Server seine aktuelle GPS-Position und das Bewegungsmuster des Benutzers. Ist der Nutzer zu Fuß unterwegs, werden die zu übermittelnden Inhalte auf Grundlage einer Umgebungssuche bestimmt. Ist er mit erhöhter Geschwindigkeit unterwegs, wird assoziiert, dass er sich in einer S-Bahn befindet. Die entsprechende Linie (z.B. S41 oder S42) wird durch die eindeutige Bewegungsrichtung des Nutzers identifiziert. Um dies zu realisieren, wird zusätzlich auf die vorherige Positionen des Nutzers zugegriffen. Daraufhin bestimmt der Server die in unmittelbarer Nähe des Clients befindlichen interessanten Orte und stellt Informationen über diese als XML-Datei dem Client zur Verfügung. Im Falle der Umgebungssuche, handelt es sich dabei um Orte, die sich im Umkreis des Nutzers befinden. Bei der anderen dahingegen, sind es die nächsten Stationen auf der aktuellen Linie der S-Bahn.

Ein Ort von Interesse wird als Point of Interest (POI) bezeichnet. Diesem sind GPS-Koordinaten, eine Beschreibung, ein oder mehrere multimediale Informationen zugeordnet. Zusätzlich kann solch ein POI einer Route zugeteilt werden. Diese Informationen werden in der MySQL Datenbank abgelegt und über das Hibernate Framework dem Server zugänglich gemacht.

Die Kommunikation zwischen Server und Client wurde nach dem Representational State Transfer (REST) Prinzip realisiert. Der Client kann über folgende URIs die REST Abfrage starten:

Ist der Client zu Fuß unterwegs: `http://server/list?latitude=..&longitude=..`

Ist der Client mit der Bahn unterwegs: `http://server/list?route=..&station=..`

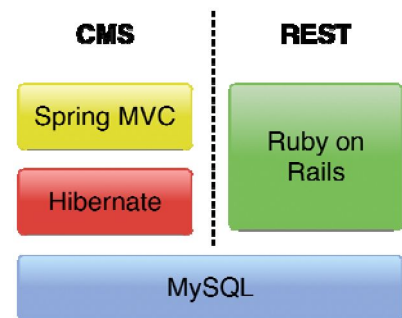


Abbildung 3: CMS Rest

Als Antwort erhält der Client eine Auflistung der für ihn interessanten POIs, inklusive der entsprechenden URL für den Download der Medien. Zudem kann über eine weitere REST Abfrage eine Liste von Kategorien bezogen werden, die es ermöglicht, die erwünschten Informationen zusätzlich zu klassifizieren und so nur Informationen zu Bereichen wie Sport, Soziales oder Kultur

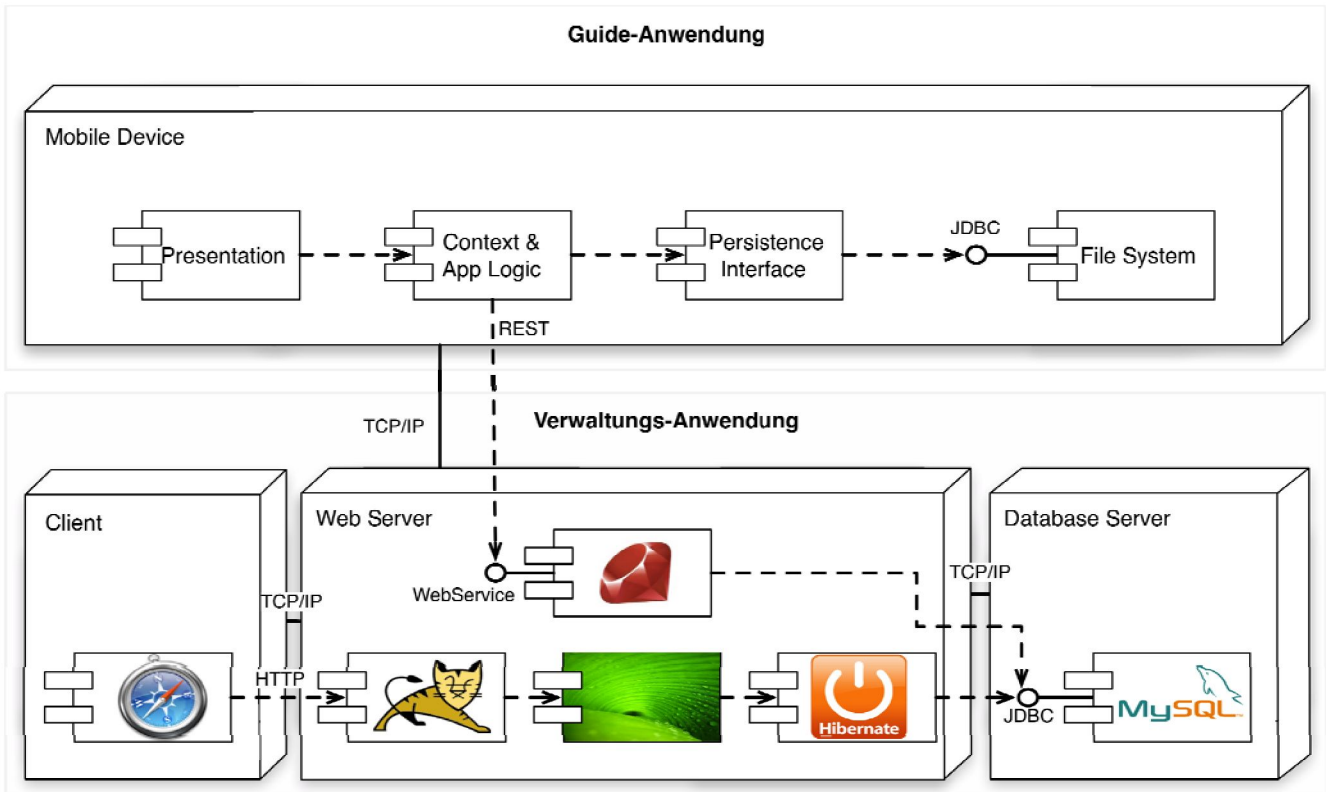


Abbildung 5: Systementwurf

zu erhalten.

Um die Skalierbarkeit des realisierten Gesamtsystems zu gewährleisten und bei den weiteren Entwicklungen die Integration weiterer Clientsysteme realisieren zu können, wurde die Nutzung von Webservices, in diesem Falle wieder das REST Prinzip, genutzt. Dies erlaubt es auf sehr einfache Weise, das Projekt weiterzuentwickeln und neue Funktionen zu integrieren.

### Positionsbestimmung und Entfernungsberechnung

Die Bestimmung und Nutzung der Position eines Nutzers sind die zentralen Komponenten der gesamten Entwicklung. Da bei jeder Client-Anfrage eine Entfernungsberechnung und eine Datenbankabfrage notwendig ist, muss die Entfernungsberechnung sehr effizient sein. Die Position wird durch die Bestimmung der Geografische Länge (engl.: Latitude) und Breite (engl.: Longitude) berechnet. Die Latitude beschreibt den Winkel, der sich zwischen Erdmittelpunkt, dem gesuchten POI und dem Äquator aufspannt. Die Longitude beschreibt den Winkel, der sich zwischen Erdmittelpunkt, dem gesuchten Punkt und dem Nullmeridian<sup>1</sup> ergibt.

Der einfachste und wohl effizienteste Ansatz ist der „Satz des Pythagoras“:

$$Distanz = \sqrt{dx^2 + dy^2}$$

$$lat = \frac{lat_1 + lat_2}{2 * \frac{\pi}{180}}$$

<sup>1</sup> Ein Meridian führt durch Nord-/Südpol. Der Nullmeridian verläuft durch die alte Greenwicher Sternwarte

$$dy = 111.3 * (lat_1 - lat_2)$$

$$dx = 111.3 * \cos(lat) * (lon_1 - lon_2)$$

$lat_2$  = Latitude Position POI

$lat_1$  = Latitude Position Client

Die Konstante 111.3 ist der Abstand zwischen zwei Breitenkreisen, angegeben in Kilometer, und  $111.3 * \cos(lat)$  der Abstand zwischen zwei Längenkreisen.

Dieser Ansatz ist bei größeren Entfernungen jedoch nicht genau genug, da die kugelförmige Gestalt der Erde vernachlässigt wird. Bei kürzeren Entfernungen sind die auftretenden Fehler zu vernachlässigen, bei großen jedoch leider nicht.

Um exaktere Werte zu erhalten kann auf den Seitenkosinussatz zurückgegriffen werden.

$$Distanz = 6378.388 * \arccos(\sin(lat_1) * \sin(lat_2) + \cos(lat_1) * \cos(lat_2) * \cos(lon_2 - lon_1))$$

Die Konstante 6378.388 entspricht dem Erdradius.

Da in der entwickelten Applikation nur Distanzen von ca. 5 bis 10 km berechnet werden, kann die vereinfachte Berechnung auf Basis des Satzes von Pythagoras genutzt werden.

### Zusammenfassung und Ausblick

Mit dem hier entstandenen Lösungspaket ist es uns gelungen, ein System zu schaffen, welches dem Benutzer einen automatischen und kontextsensitive Informationsfluss bereitstellt. Besonderes Augenmerk wurde bei der Umsetzung des Clients auf ein barrierearmes User Interface gelegt. Die barrierearme Aufbereitung, speziell die nonvisuellen Bedienbarkeit der Anwendung und der beschreibende Produktionsstil der auditiven Inhalte machen den Carpe Guide besonders für Blinde und Sehbehinderte interessant.

Da dieser Multimedia Guide als Multilinguales System entwickelt wurde, kann es ohne Probleme in jede beliebige Sprache portiert werden. Es müssen lediglich die Inhalte in der jeweiligen Sprache bereitgestellt werden. Um die Führungen weiter zu personalisieren, ist es möglich, weitere Kategorien zu erstellen, damit der Client diese dem Nutzer verfügbar und auswählbar machen kann. Somit wird die multimediale Tour nach dem Wunsch des Nutzer individuell gestaltet und er erhält nur Informationen für die er sich interessiert.

Eine Portierung des Carpe Guide in andere Städte und Länder ist dank der genutzten Lokalisierungsmethodik und mit dem Einpflegen der jeweiligen neuen multimedialen Inhalte sehr einfach umzusetzen.

Damit die Barrierefreiheit weiter verfeinert werden kann, bietet sich eine Erweiterung des Multimedia Guides an, um zum Beispiel die Gebärdensprache über Videos zu unterstützen oder dem blinden und sehbehinderten Nutzer den visuellen Content auditiv aufzubereiten, wodurch auch die rein visuellen Informationen, wie zum Beispiel die aktuelle Route oder die nächste Station, einem Blinden zugänglich wäre.

### Weiterführende Informationen

- [1] Research group "Information and Communication Systems", University of Applied Sciences (HTW Berlin), <http://inka.f4.htw-berlin.de>.
- [2] Inc., Apple: Cocoa Fundamentals Guide: The Model-View-Controller Design Pattern, <http://developer.apple.com/documentation>.
- [3] Inc., Apple: Apple Developer Connection – Developer Tools and Technologies Download Xcode. <http://developer.apple.com/technology/xcode.html>.
- [4] J.Schiller, Mobilkommunikation, Addison-Wesley, 2003.  
<http://inka.f4.htw-berlin.de>
- [5] Carpe Berlin, <http://www.carpeberlin.de>.