

# WayFinder

Entwicklung einer Navigationsplattform für die  
dynamische Fußgängerführung

Borys Kogan

26. Februar 2008

## Inhaltsverzeichnis

<b>1</b>	<b>Abstrakt</b>	<b>4</b>
<b>2</b>	<b>Einleitung</b>	<b>5</b>
<b>3</b>	<b>Komponentenarchitektur</b>	<b>6</b>
<b>4</b>	<b>Konzeption</b>	<b>8</b>
4.1	AStar . . . . .	8
4.2	Graphenrepräsentation der GEO-Objekte . . . . .	9
4.3	Points of Interest . . . . .	9
4.4	Kontextabhängige Routenerstellung . . . . .	10
4.5	Graphische Darstellung der GEO-Objekte . . . . .	11
4.5.1	Bildrepräsentation . . . . .	11
4.5.2	City GML . . . . .	11
<b>5</b>	<b>Realisierte Funktionalität</b>	<b>14</b>
<b>6</b>	<b>Die Technologienübersicht</b>	<b>16</b>
<b>7</b>	<b>Fazit</b>	<b>17</b>
<b>8</b>	<b>Referenzen</b>	<b>18</b>

## Abbildungsverzeichnis

1	Architektur der beteiligten Anwendungskomponenten . . . . .	6
2	Grapherepräsentation der Räume und möglicher Wege zwischen denen	9
3	Statische graphische Darstellung der Gebäudekarte . . . . .	12
4	Ausschnitt aus der CityGML-Beschreibung der 11.Etage der HAW Ham- burg . . . . .	13
5	Auswahl zwischen den Repräsentationsmodellen . . . . .	14
6	Darstellung der berechneten Route aus dem Knoten 54 zu dem Knoten 130 (graphisch und optional als eine Liste der passierten Knoten) . . .	15
7	Optionale Eingabe der POIs . . . . .	15
8	Optionale Eingabe eines Kontextes . . . . .	15

# 1 Abstrakt

## WayFinder

Entwicklung einer Navigationsplattform für die dynamische Fußgängerführung

## Stichworte

Route, Suchalgorithmus, Innenraum Navigationssystem, CityGML, Navigation, Kontextabhängige Suche.

## Abstrakt

Diese schriftliche Ausarbeitung beschreibt die Grundfunktionalitäten und die für die Implementierung benutzten Technologien der Anwendung, die im Rahmen des Projektes entwickelt wurde. Als Hauptziel der Plattform kann man die kontextabhängige Modellierung der Route nennen.

## 2 Einleitung

Über die Vision eines innovativen Navigationssystems für Fußgänger wurde auf den Seiten meiner Ausarbeitungen zu den Anwendungen I und II diskutiert. Dem Leser wurde ein Konzept vorgestellt, wie man in einem unbekanntem und unübersichtlichen Gebäude schnell und effizient zu einem oder zu mehreren Zielen ohne zusätzliche Hardware-Ausstattung navigiert werden kann. Als Hilfsmittel für die Darstellung der Navigationshinweise wurde die dynamische Beschilderung vorgeschlagen, die heutzutage in jedem Flughafen oder z.B. in jeder Messehalle installiert ist.

Auf den Seiten der vorliegenden Ausarbeitung findet der Leser eine Beschreibung des im Rahmen des Master-Projektes entwickelten Software-Produktes. Die Anwendung soll als Basis für das zukünftige Indoor-Navigationssystem dienen, um folgende Funktionalität zu realisieren:

- Berechnung der Route
- Betrachtung der möglichen Zwischenziele
- dynamische Setzung der Prioritäten entsprechend dem Kontext
- graphische Darstellung der Gebäude-Modelle

Dabei bewegt man sich auf der so genannten administrativen Ebene, die für den Endbenutzer nicht sichtbar ist. Es ist aber eine unabdingbare Kalkulationskomponente des kompletten Navigationssystems.

Als die Simulationsumgebung wurde die 11. Etage der HAW Hamburg genommen. Aus der existierenden technischen Raumdarstellung wurde die passende graphische und XML-basierende Repräsentation der Etage entwickelt. So bekommt man eine Plattform, auf deren Basis verschiedene Beispielszenarien in der Indoor-Navigation-Metapher implementiert werden können. Ein mögliches Szenario ist in [Lit 1] beschrieben.

### 3 Komponentenarchitektur

In diesem Kapitel hat der Leser eine Möglichkeit, die Bestandteile der Anwendung kennen zu lernen. Das Zusammenspiel der Komponenten wird auf Basis der Architektur vorgestellt. Für die Lösung der oben beschriebenen Aufgaben des Systems sind die auf der Abbildung 1 dargestellten "Bausteine" notwendig.

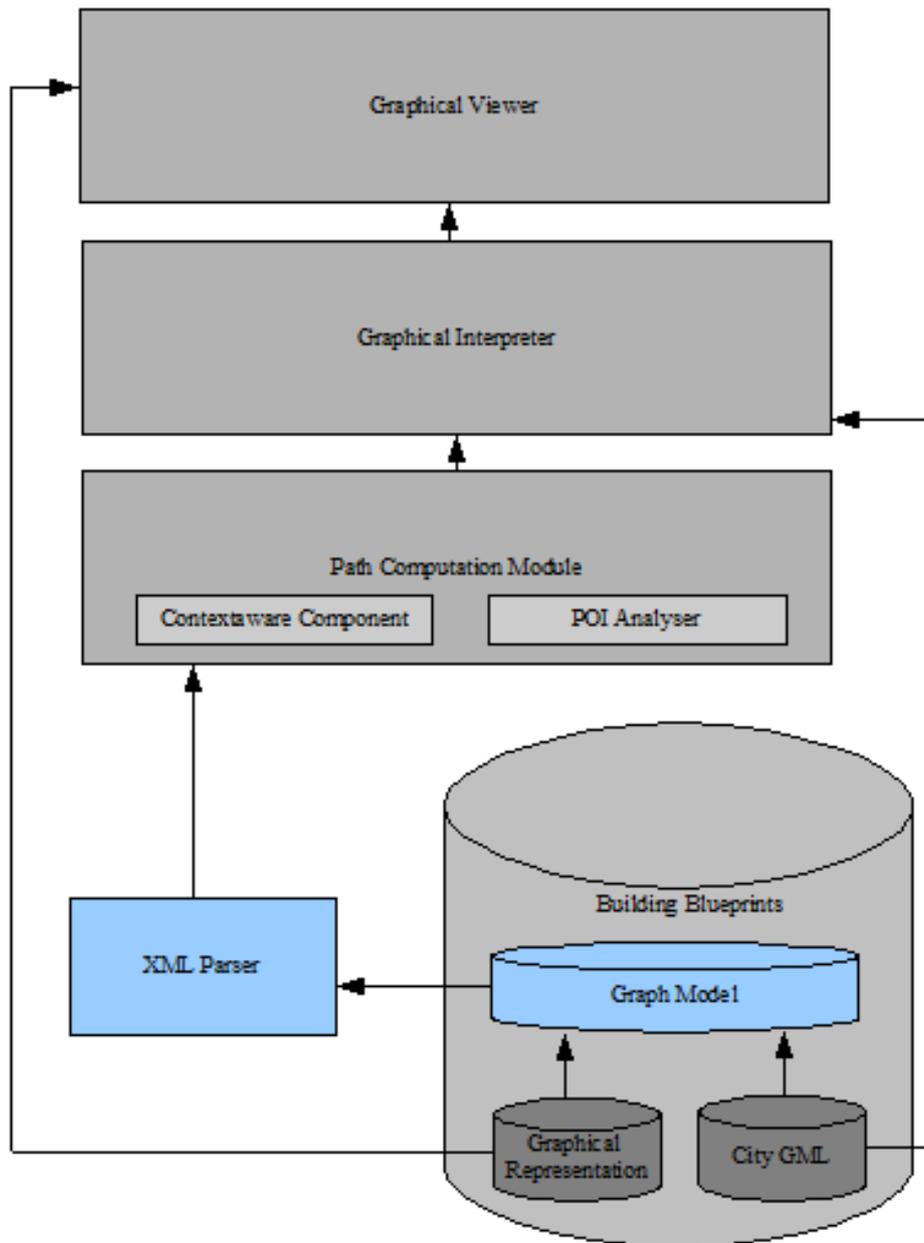


Abbildung 1: Architektur der beteiligten Anwendungskomponenten

Beschreibung der Komponentenfunktionalität:

- Building Blueprints

Der Datenbestand enthält die Basis-Modelle des Gebäudes. Der *WayFinder* bietet 2 Möglichkeiten, das Gebäude für die weitere Berechnung der Route zu beschreiben. Die weitere benötigte Schicht ist eine Graphenrepräsentation der GEO-Objekte.

1. Graphical Representation

Ist eine statische Bildrepräsentation des Gebäudes.

2. CityGML

Eine XML-Darstellung des Gebäudes.

3. Graph Modell

Eine Kombination von Knoten und Kanten, die die Räume, Türen des Gebäudes und die möglichen Wege zwischen ihnen beschreiben. Wird für die Routenberechnung benutzt.

- XML Parser

Dies ist eine Hilfskomponente, die aus dem XML-basierten Graphenmodell die für die weitere Routenberechnung benötigten geometrischen Daten extrahiert und weitergibt.

- Path Computation Module

Enthält den eigentlichen Suchalgorithmus. Erstellt die Route nach den Präferenzen des Benutzers.

- Graphical Interpreter (graphischer Interpreter)

Ist dafür zuständig, um die gelieferten Rohdaten für die weitere graphische Abbildung zu interpretieren. Die Eingabedaten werden von dem *Path Computation Module* für die weitere Abbildung der gefundenen Route und vom *CityGML* für die graphische Repräsentation der Gebäudekarte zur Verfügung gestellt.

- Graphical Viewer

Interaktive graphische Darstellung der Suchergebnisse und der Gebäudekarte.

## 4 Konzeption

Im Abschnitt 2 wurden dem Leser die Hauptaufgaben der implementierten Anwendung aufgelistet, im Kapitel 3 wurden die dafür zuständigen Komponenten vorgestellt und beschrieben. In diesem Abschnitt der Ausarbeitung tauchen die oben beschriebenen Anforderungen und Bestandteile der Anwendung wieder auf, aber mit einer detaillierten Beschreibung der dahinter stehenden Funktionalität und Realisierungsalgorithmen. Zu einigen benutzten Technologien wird auch eine kurze Einleitung aus den theoretischen Grundlagen zur Verfügung gestellt.

### 4.1 AStar

Eine Basisaufgabe jedes Navigationssystems ist natürlich eine effiziente Suche des schnellsten oder besser gesagt des effizientesten Weges vom Start- zum Zielpunkt. Warum sind die Charakteristiken schnell und effizient in diesem Fall unterschiedlich? Der Weg, der dem Fußgänger angeboten wird, muss natürlich den eingegebenen Präferenzen des Benutzers entsprechen. Sei es z. B. eine Reihe von Zwischenzielen, die in der Endroute enthalten sein müssen oder der Einfluss des Kontextes (nicht vorgesehene Ereignisse). *AStar* ist dafür zuständig, diese Berechnungen durchzuführen und die Route zu modellieren. Es ist eine Implementierung des bekannten  $A^*$  - Suchalgorithmus.

**Definition 1 (Aus Wikipedia)** *Der  $A^*$ -Algorithmus („A Stern“ oder englisch „a star“) gehört zur Klasse der informierten Suchalgorithmen. Er dient in der Informatik der Berechnung eines kürzesten Pfades zwischen zwei Knoten in einem Graphen mit positiven Kantengewichten. Er wurde das erste Mal 1968 von Peter Hart, Nils Nilsson und Bertram Raphael beschrieben. Im Gegensatz zu uninformierten Suchalgorithmen verwendet der  $A^*$ -Algorithmus eine Schätzfunktion (Heuristik), um zielgerichtet zu suchen und damit die Laufzeit zu verringern. Der Algorithmus ist optimal. Das heißt, es wird immer die optimale Lösung gefunden, falls eine existiert.*

*AStar* erfüllt folgende Funktionalität. Die Graphenrepräsentation der Gebäudekarte wird geladen und geparkt. Die Ebene des Gebäudes wird als eine zusammenhängende Menge von Knoten und Kanten dargestellt. Als Startposition nimmt man den nächstliegenden Knoten innerhalb des Raumes und definiert das gesuchte Ziel. Mit Hilfe des Suchalgorithmus wird die schnellste Route definiert und danach graphisch abgebildet. Der weitere Abschnitt betrachtet näher die Graphenrepräsentation des Gebäudes.

## 4.2 Graphenrepräsentation der GEO-Objekte

Um den Suchalgorithmus zu 'aktivieren', muss das Gebäude (oder ein Teil davon) in der passenden Form repräsentiert werden. Deswegen wurde die ausgewählte Simulationsumgebung (11. Etage der HAW Hamburg) als Kombination von Knoten und Kanten beschrieben. Knoten repräsentieren dabei Räume der Etage, Kanten die möglichen Wege zwischen ihnen. Die Beschreibung wurde als eine XML-Datei abgelegt. Daraus wird ein Graph der Etage generiert und als Datenbasis für den Suchalgorithmus benutzt. Die Topologie der platzierten Knoten ist nach dem folgenden Prinzip entstanden: Knoten befinden sich in der Mitte eines Raumes, in der Tür, gegenüber der Tür, falls es um einen langen unübersichtlichen Raum ohne Entscheidungspunkte(z. B. Korridor) handelt(s.Abb. 2).

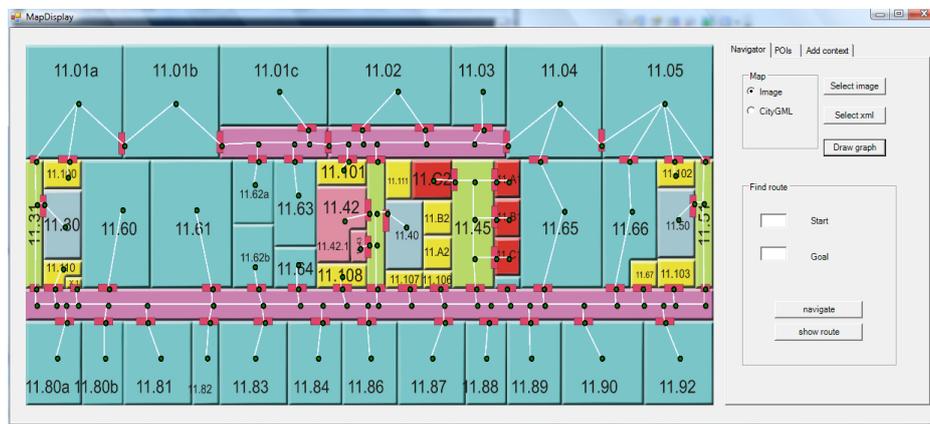


Abbildung 2: Graphrepräsentation der Räume und möglicher Wege zwischen denen

## 4.3 Points of Interest

Eine zusätzliche Option, die in dem *WayFinder* implementiert wurde ist das Konzept der *Points-Of-Interest* (weiter POI genannt). In der realen Welt ist es schon eine bekannte Tatsache, dass für einen sich bewegenden Fußgänger Navigation ein Zwischenziel ist. Das heißt, wir dürfen die Anforderungen eines Benutzers nicht als eine Befehlsreihenfolge Start  $\rightarrow$  Finish interpretieren, sondern wir müssen davon ausgehen, dass für den Benutzer verschiedene Zwischenziele von Interesse sein können, die auf dem Weg zum Endziel besucht werden müssen.

In dem *WayFinder* können verschiedene POIs (maximal 5) eingegeben werden. Die Endroute besteht dann aus der Reihe von Zwischenrouten, die die eingegebenen Objekte miteinander verbinden. Der entwickelte Algorithmus erlaubt es, aus allen möglichen Varianten nach der optimalen(kürzesten) Route zu suchen.

#### 4.4 Kontextabhängige Routenerstellung

Über den Kontext, in dem sich der Fußgänger bewegt wurde in [Lit 2] und [Lit 1] diskutiert. In meiner Vision des Navigationssystems gehe ich davon aus, dass das Tracking der Benutzer nicht nötig und nicht als Teil der geplanten Arbeit zu betrachten ist. Deswegen muss dem System der vollwertige Datenbestand zur Verfügung gestellt werden, damit die reale Welt und die möglichen Verhaltensmodelle der Umgebung vorhersehbar sind. In der hier vorgestellten Anwendung wurde ein Mechanismus eingebaut, mit dessen Hilfe man den Kontext eingeben kann. Dementsprechend ändern sich die Suchergebnisse der *AStar*-Suchkomponente.

Wie es schon auf den Seiten dieser Ausarbeitung genannt wurde, läuft die Suche der optimalen Route auf der Graphenrepräsentation des Gebäudes. Dabei handelt es sich um einen gewichteten und gerichteten Graphen. So bekommt man die Möglichkeit, die Gewichtung zu manipulieren. Das bedeutet, dass die Zugänglichkeit zu den Räumen kontrolliert bzw. gesperrt werden kann. In dem *WayFinder* sind 2 verschiedene Kontext-Szenarien eingebaut:

- Extraordinary event (außergewöhnliches Ereignis)  
Stellt eine mögliche sicherheitskritische Situation (z. B. Brandfall) dar. Eine Reihe von Räumen müssen dabei blockiert werden und dem Benutzer wird eine alternative Route angeboten, obwohl sie nicht unbedingt die schnellste ist. In der Anwendung wird es dadurch realisiert, dass die Gewichtung der betroffenen Räume als eine unendlich Große Zahl dargestellt wird und der Suchalgorithmus betrachtet alle Routen, die den zu blockierenden Raum enthalten, als nicht effizient.
- Capacity overload (Raumüberlastung)  
Ein Beispiel aus der realen Welt: die gerade gelandete Gruppe von Touristen aus Frankreich folgt jetzt zu den Laufbändern für Gepäckabholung. Die Auslastung der Räume, durch die die Gruppe geht, wird automatisch erhöht, damit kein Stau verursacht wird, falls die Räume von mehreren Gruppen benutzt werden sollen. So wird der zweiten Gruppe, die auf die reservierten Ressourcen zugreifen möchte, eine alternative Route angeboten, falls vorhanden. Die Gewichtung der Kanten wird erhöht aber die Beträge hängen von den Bearbeitungskapazitäten der Räume ab.

## 4.5 Graphische Darstellung der GEO-Objekte

Ein wichtiger Aspekt der Realisierung eines Navigationssystems ist die Repräsentation der Ergebnisse für den Endbenutzer. Da die vorgestellte Anwendung in erster Reihe für die administrative und Testzwecke geeignet ist, wird das Konzept der dynamischen Beschilderung, die als Abbildungsmedium für die Navigationshinweise vorgesehen ist, in dieser Arbeit nicht betrachtet. Stattdessen stellen sich die Fragen, wie die GEO-Information für den Entwickler dargestellt werden kann, in welche besser geeignete Form als technische Zeichnungen die Karten der Gebäude umgewandelt werden können. Welche Technologien sind dafür am besten geeignet? Das Hauptkriterium bei der Überlegung war folgendes: nicht nur die geometrische Repräsentation der Raumobjekte sondern auch die semantische Bedeutung für die effiziente Indoor-Navigation notwendig ist. Im *WayFinder* sind dem Entwickler 2 verschiedene Möglichkeiten angeboten.

### 4.5.1 Bildrepräsentation

Dafür wurde der vorhandene Gebäudeplan(PDF) der 11.Etage von der HAW manuell in eine Bildrepräsentation(JPG) umgewandelt. Folgende Objekte wurden in die graphische Darstellung übernommen: Lehrräume, Korridore, Toiletten, Nebenräume. Auf die Dichte der Wände, Kolonnen, Scheidewände, Raumausstattung wurde verzichtet. Als ein wesentlicher Vorteil der graphischen Raumrepräsentation kann man die offensichtliche Einfachheit der Erstellung nennen. Die semantische Unterscheidung der Räume voneinander ist natürlich nicht möglich, weil die graphische Repräsentation statisch ist und beinhaltet keine Information über die Meta-Objekte. Falls man eine semantische Bedeutung der Objekte braucht, dann muss diese zusätzlich in das Graphenmodell des Gebäudes eingetragen werden, z. B. als zusätzliche Eigenschaften der Räume. Die weitere Abbildung zeigt das entworfene graphische Modell der Gebäudekarte (s. Abb. 3).

### 4.5.2 City GML

Eine elegantere Lösung bietet der neu entwickelte Standard *CityGML*, der sich heutzutage als eine mächtige Technologie für die Beschreibung der Urban-Objekte bewährt hat.

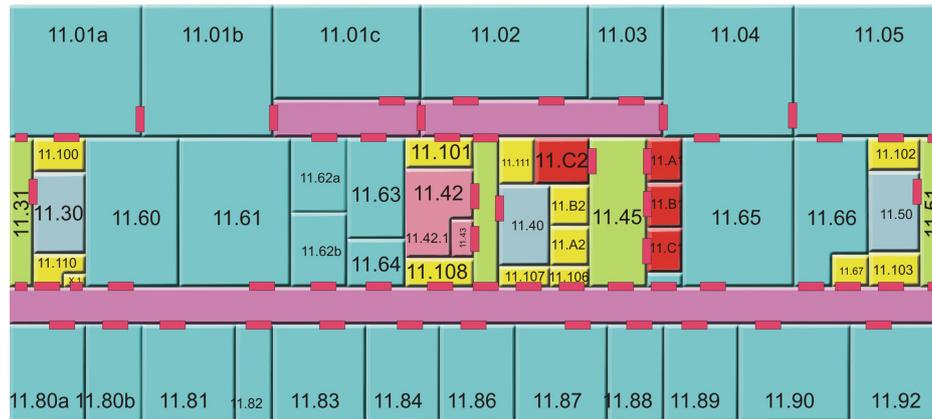


Abbildung 3: Statische graphische Darstellung der Gebäudekarte

**Definition 2 (Aus der offiziellen CityGML-Seite)** *CityGML is a common information model for the representation of 3D urban objects. It defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical and appearance properties. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. These thematic information go beyond graphic exchange formats and allow to employ virtual 3D city models for sophisticated analysis tasks in different application domains like simulations, urban data mining, facility management, and thematic inquiries.*

Die Objektbeschreibung ist nichts anderes als eine XML-Datei, die nach einem vordefinierten Schema erstellt wird. Für den *WayFinder* wurde eine prototypische Gebäudekarte modelliert, die den Regeln der *CityGML* Standardisierung entspricht. Auf der offiziellen Seite der *CityGML*-Entwickler ist eine Reihe von Hilfstools angeboten. Leider wurde kein Parser für die .NET-Plattform implementiert, auf der die vorgestellte Anwendung *WayFinder* realisiert ist. Folgende Komponenten wurden für die Analyse und Bearbeitung von *CityGML*-Karten im Rahmen des Master-Projektes implementiert:

- XML Validator  
Überprüfung der Datei nach dem vorgegebenen XSD-Schema (vermeidet die mögliche Inkompatibilität mit dem offiziellen Standard).
- CityGML Parser  
Die Komponente liest die XML-Darstellung des Gebäudes und gewinnt daraus die geometrische Information über die einzelnen Räume.
- Graphischer Interpretier

Die Komponente stellt eine graphische Interpretation der vorher abgelesenen Raumobjekte dar.

Die einzelnen Räume sind als Polygone beschrieben. Da es sich um eine prototypische XML-Beschreibung handelt, wurde die Semantik noch nicht integriert. Die folgende Abbildung (Abb. 4) zeigt einen Ausschnitt aus der CityGML-Karte, auf dem die hierarchische Beschreibung des Raumes 11.110 zu sehen ist.

```
<interiorRoom>
  <Room gml:id="R11.110">
    <lod4MultiSurface>
      <gml:MultiSurface>
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos srsDimension="2">55 610</gml:pos>
                <gml:pos srsDimension="2">185 610</gml:pos>
                <gml:pos srsDimension="2">185 660</gml:pos>
                <gml:pos srsDimension="2">130 660</gml:pos>
                <gml:pos srsDimension="2">130 695</gml:pos>
                <gml:pos srsDimension="2">55 695</gml:pos>
                <gml:pos srsDimension="2">55 610</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:MultiSurface>
    </lod4MultiSurface>
  </Room>
</interiorRoom>
```

Abbildung 4: Ausschnitt aus der CityGML-Beschreibung der 11.Etage der HAW Hamburg

## 5 Realisierte Funktionalität

In diesem Abschnitt werden dem Leser die aktuell verfügbaren und die im Rahmen des Projektes implementierten Systemfunktionalitäten mit Hilfe der Programm-Screenshots und der Beschreibung präsentiert. So bekommt man eine Vorstellung davon, wie der Programmablauf in der Realität aussieht.

Nachdem die Anwendung gestartet wurde, hat der Benutzer die Auswahlmöglichkeit, die Gebäudekarte entweder als eine *CityGML* oder eine Bilddatei zu laden (s. Abb. 5). Auf den vorherigen Abbildungen wurde dem Leser eine Anwendung mit den geladenen JPG-Repräsentationen der Gebäudeebene vorgestellt (s. Abb. 3).

Die Karte des Gebäudes wurde geladen (im Falle der *CityGML*-Darstellung geparkt und

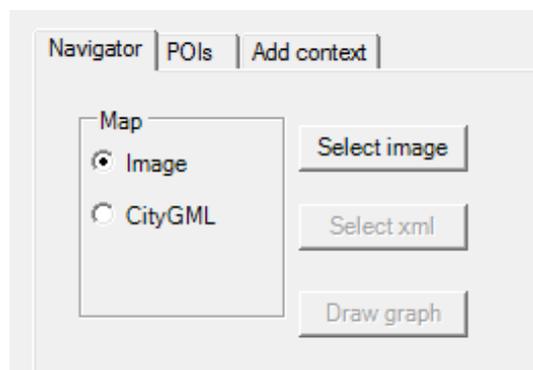


Abbildung 5: Auswahl zwischen den Repräsentationsmodellen

graphisch dargestellt). Die nächste Schicht ist die Graphendarstellung der möglichen Wege zwischen den Räumen, die als eine separate XML-Datei abgelegt ist. Die Graphenrepräsentation wird geladen. Eine vollständige Variante des Paares Karte-Graph sieht man auf der Abbildung 2.

Nun hat der Benutzer die Möglichkeit, die Route von einem Startpunkt zu dem ausgewählten Zielpunkt berechnen zu lassen. Dafür werden die entsprechenden Textfelder aktiviert und die ausgewählten Knoten werden als Start und Ziel eingetragen (s. Abb. 6). Der Annäherungsalgorithmus sorgt dafür, dass die am dichtesten liegenden Knoten automatisch identifiziert werden. Eine weitere Option ist, eine Reihe von Zwischenzielen (Points of Interest) einzugeben. Das wird nach dem gleichen Prinzip, wie die Auswahl der Start-, Endknoten realisiert (s. Abbildung 7) Die Möglichkeit, die Routenerstellung von einem Kontext beeinflussen zu lassen wird dadurch realisiert, dass der Benutzer zunächst den getroffenen Knoten eingibt und dafür einen der beiden Kontexte setzt (s. Abb. 8). Falls der eingegebene Knoten in der Route auftauchen soll, wird der Algorithmus nach den alternativen Wege suchen.

## 5 REALISIERTE FUNKTIONALITÄT

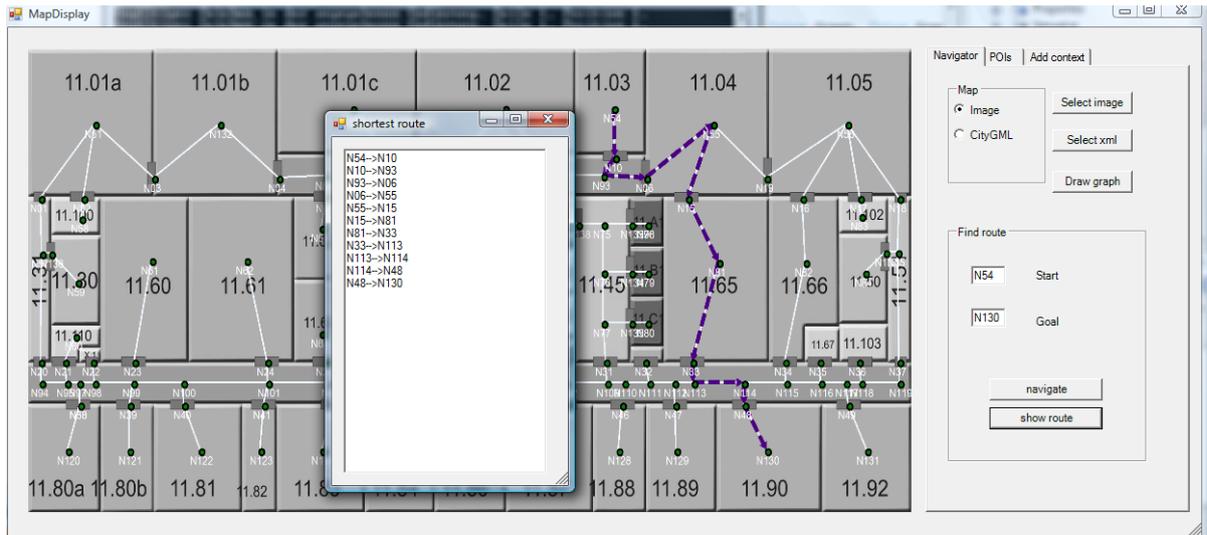


Abbildung 6: Darstellung der berechneten Route aus dem Knoten 54 zu dem Knoten 130 (graphisch und optional als eine Liste der passierten Knoten)

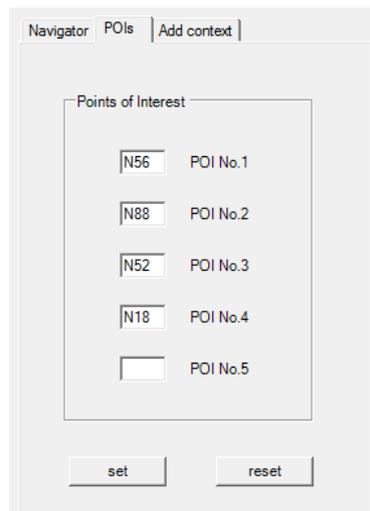


Abbildung 7: Optionale Eingabe der POIs

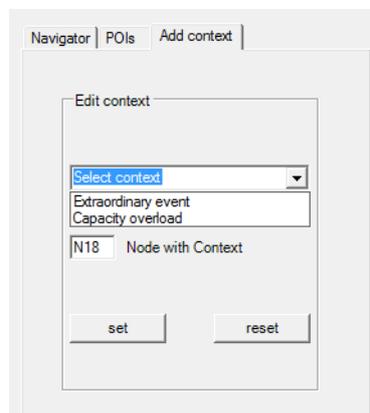


Abbildung 8: Optionale Eingabe eines Kontextes

## 6 Die Technologienübersicht

In diesem Abschnitt werde ich die für die Implementierung der *WayFinder*-Anwendung eingesetzten Technologien und die dafür entscheidenden Punkte nennen.

- Programmiersprache

Es wurde schon gesagt, dass die entwickelte Anwendung für die möglichen Testzwecke benutzt werden sollte. Das heißt, dass man bei der Navigation der Fußgänger für die Analyse der Ergebnisse des geplanten Navigationssystems auf Basis der dynamischen Beschilderung ein Tool braucht, das auf den mobilen Geräten lauffähig ist (wegen des Trackings der Fußgänger) und eventuell mit dem EMIC Location and Mapping Framework für die Weiterentwicklung kompatibel sein muss. Die ausgewählte Lösung ist C#.

- Entwicklungsumgebung

Microsoft Visual Studio 2005, das eine breite Palette von Hilfsmechanismen bei der Erstellung der C#-Anwendungen anbietet.

- XML

Ein weit verbreitetes Format für den Datenaustausch und für die hierarchisch strukturierte Objektbeschreibung. In der Anwendung wird eine selbstentwickelte Form des XML-Dokumentes benutzt, um die Graphenrepräsentation des Gebäudes zu beschreiben.

- CityGML

Ein mächtiger Mechanismus, um die Stadtobjekte zu beschreiben. Wird als eine der beiden Möglichkeiten (CityGML vs. Bilddarstellung) für die Darstellung der Gebäudekarten benutzt.

- Graphentheorie

Theoretische Grundlagen, um die geometrischen Daten der Gebäudeinfrastruktur für die Suchalgorithmen bearbeitbar zu machen.

- A\*

Eine der besten heuristischen Suchen, wird für die Ermittlung der Route vom Start- zu einem Zielpunkt benutzt.

## 7 Fazit

Die für das Projekt gesetzten Ziele und die als Ergebnis realisierte Anwendung sind ein hilfreicher Schritt in der Vorbereitungsphase für die zukünftige Masterarbeit. Es wurde ein Versuch gemacht, eine Plattform zu entwickeln, die als Basis für das geplante Fußgängernavigationssystem dienen soll. Das Produkt, das im Rahmen des Projektes entstanden ist, kann man folgendermaßen charakterisieren:

Eine Standalone-Anwendung für die kontextabhängige Berechnung der Route von einem Startpunkt zu dem ausgewählten Ziel mit der optionalen Möglichkeit, Points of Interest als Zwischenziele in die Suchpräferenzen einzutragen. Die intensive Auseinandersetzung mit den Grundlagen von *CityGML* Standard erlaubten es, eine zusätzliche Option für die Darstellung der Gebäudeobjekte in die Anwendung zu integrieren. Im Rahmen des Projektes hat eine intensive Kommunikation und Zusammenarbeit mit den anderen Projektbeteiligten und Mitkommilitonen stattgefunden, mit deren Hilfe einige innovative Lösungsansätze implementiert wurden. Für die Integrierung der implementierten Anwendung in die Realisierungsphase der geplanten Masterarbeit sind folgende Aspekte zu verbessern:

- *CityGML* Darstellung der Simulationsumgebung (11. Etage der HAW Hamburg) muss vervollständigt werden und die semantische Beschreibung der Raumobjekte ist gewünscht.
- Die Kontextabhängige Suche wird um die stochastischen Modelle erweitert, um die Auslastungskontrolle der Räume zu realisieren.
- Eine zweidimensionale Darstellung des Gebäudes soll um die dritte Dimension erweitert werden, damit die Suche auf mehreren Ebene möglich ist.

Die erreichte Funktionalität der Anwendung kann für verschiedene Zwecke bei der Erstellung eines Navigationssystems durch die angebotenen Schnittstellen benutzt werden.

## 8 Referenzen

### Literatur

- [Lit 1] DYNAMISCHE BESCHILDERUNG IN DER FLUGHAFEN-METAPHER  
Borys Kogan, HAW Hamburg, Ausarbeitung im Rahmen der Anwendung I,  
Juli 2007
- [Lit 2] DYNAMISCHE BESCHILDERUNG IN DER FLUGHAFEN-METAPHER(TEIL II)  
Borys Kogan, HAW Hamburg, Ausarbeitung im Rahmen der Anwendung II,  
Februar 2008
- [Lit 3] GRAPHENTHEORIE 2.AUFLAGE  
R.Diestel, 2000
- [Lit 4] CITYGML  
Exchange and Storage of Virtual 3D City Models,  
www.citygml.org, 2008
- [Lit 5] C# Die neue Sprache für Microsofts .NET-Plattform,  
Eric Gunnerson
- [Lit 6] XML - Insider,  
Markt u. Technik, München (Juli/August 2001)
- [Lit 7] Alewtina Schumann,  
AW I Ausarbeitung,  
WS 2007
- [Lit 8] A Semantic Indoor Navigation System  
Christos Anagnostopoulos, Vassileios Tsetsos, Panayotis Kikiras  
Pervasive Computing Research Group, Communication Networks Laboratory,  
Dept of Informatics & Telecommunications, University of Athens,  
Panepistimioupolis, Athens 15784, Greece
- [Lit 9] EMIC Location and Mapping Framework,  
<http://www.microsoft.com/emea/emic/downloadcentre.mspx>.  
2007