

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Information Systems for Public Transport Users

Pedro Ferreira

Master in Informatics and Computing Engineering

Supervisor: Rosaldo Rossetti (Auxiliar Professor)

Co-Supervisor: António Coelho (Auxiliar Professor)

28th June, 2010

Information Systems for Public Transport Users

Pedro Ferreira

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: António Augusto de Sousa (Associate Professor)

External Examiner: Elisabete Maria Mourinho Arsénio Guterres de Almeida (Auxiliar Investigator - LNEC)

Supervisor: Rosaldo José Fernandes Rossetti (Auxiliar Professor)

31st July, 2010

Abstract

Until now, the existing Geographical Information Systems were mainly developed concerning the needs of vehicle users. However in recent years with the proliferation and evolution of mobile devices, navigation systems started to be directed towards the needs of the pedestrian users but their full potential is still far from being exploited. Pedestrians have a higher degree of freedom so the information contained in these systems is different from that destined for vehicles. There is also the information about public transport networks which is how the pedestrian usually travels great distances.

The main goals of this dissertation are to present the specification of user-centered solution based on location with a suitable service-oriented architectural basis for Geographical Information Systems destined to pedestrians and as such, to public transport users as well; to provide a spatiotemporal data model which considers not only pedestrian navigable areas such as parks and sidewalks but gathers in itself information about the several existing public transport networks and schedules; and to provide a multimodal routing algorithm that makes use of the information contained in the data model.

Resumo

Até agora, os actuais Sistemas de Informação Geográfica foram desenvolvidos principalmente tendo em conta as necessidades dos utilizadores de veículos. No entanto, nos últimos anos com a proliferação e evolução dos dispositivos móveis, os sistemas de navegação começaram a ser direccionados para as necessidades dos utilizadores pedestres. Mas, o seu potencial ainda está muito pouco explorado. Dado que os pedestres têm um maior grau de liberdade, as informações contidas nestes sistemas são diferentes daquelas destinadas aos veículos. Há que ter também em conta a informação sobre as redes de transportes públicos que são o meio de transporte geralmente utilizado pelos pedestres para percorrer grandes distâncias.

Os principais objectivos deste trabalho são apresentar a especificação de uma solução baseada em localização centrada no utilizador com uma base arquitectural orientada a serviços adequada a Sistemas de Informação Geográfica destinados a pedestres e, como tal, para os utilizadores dos transportes públicos; fornecer um modelo de dados espaço-temporal que considere não só a informação para áreas navegáveis apenas por pedestres, como parques e passeios, mas, que reúna também em si informações e horários sobre as diversas redes existentes de transportes públicos; e, fornecer um algoritmo de routing multimodal que faça uso das informações contidas no modelo de dados.

Acknowledgements

I would like to present my thanks to my supervisor Prof. Rosaldo Rossetti and co-supervisor Prof. António Coelho for all their support during the present project and for the help given when the work of this dissertation was not going so well. Also, I would like to thank my parents and my brother for all that they have given me until now and for being there for me when I needed the most.

Pedro Ferreira

Contents

1. Introduction	1
1.1 Motivation.....	2
1.2 Problem Statement.....	3
1.3 Objectives	4
1.4 Expected Results	5
1.5 Structure of the Dissertation	6
2. State of the Art	7
2.1 Introduction	7
2.2 Requirements for pedestrians	8
2.3 Problems in Pedestrian Navigation Systems	13
2.3.1 Speed.....	13
2.3.2 Position Determination.....	13
2.3.3 Navigation aids.....	14
2.3.4 Information.....	14
2.3.5 Adaptivity of the system to the user.....	15
2.3.6 Routing.....	15
2.4 Information systems for public transport users	16
2.4.1 Existing solutions	16
2.4.2 Planning algorithm	18
2.4.3 Data model	24
2.5 Chapter Summary.....	28
3. A Location-Based Solution	30
3.1 Introduction.....	30
3.2 Methodology	31

CONTENTS

3.3 Requirements Specification.....	32
3.4 System Architecture	37
3.4.1 General Overview	37
3.4.2 Logical Architecture.....	39
3.4.3 Physical Architecture	41
3.5 Chapter Summary.....	43
4. Data Model.....	44
4.1 Introduction	44
4.2 Pedestrian Information	47
4.3 Public Transport Information	55
4.4 Additional Information.....	57
4.5 Chapter Summary	60
5. Sensor Enhanced Navigation (Magnetometer and Accelerometer).....	61
5.1 Introduction	61
5.2 Implementation	63
5.2.1 Heading Information (Magnetometer)	64
5.2.2 Acceleration information (Accelerometer)	66
5.3 Chapter summary	68
6. Conclusions	69
6.1 Final Remarks	69
6.2 Future Work	71
References	72
A: Requirements Specification for Public Transport Users.....	75
A.1 Introduction	75
A.2 Requirements.....	76
A.3 Chapter Summary.....	94

CONTENTS

List of Figures

Figure 2.1: Source: [GRO07]	9
Figure 2.2: Source: [GRO07]	10
Figure 2.3: Source: [GRO07]	11
Figure 2.4: Source: [GRO07]	12
Figure 2.5: Urban Canyon	14
Figure 2.6: Navitime Supported Transport Types.....	16
Figure 2.7: Navitime Walking Route Example	17
Figure 2.8: Google Maps Web Interface now with walking directions	18
Figure 2.9: NimbleTransit's aggregation process. Source: [CHE08]	21
Figure 2.10: Example of result from NimbleTransit composition process. Source: [CHE08] ...	22
Figure 2.11: Class diagram of the transit network. Source: [HUA08].....	25
Figure 2.12: Context and user aware spatial query schema. Source: [ZIP06].....	26
Figure 2.13: Graph describing actions necessary for a legal ride on a metro. Source: [FRA07]	27
Figure 3.1: General Overview of the Architecture.....	38
Figure 3.2: Logical Architecture	39
Figure 3.3: Profile Configurator.....	40
Figure 3.4: Physical Architecture.....	42
Figure 4.1: Data Model	46
Figure 4.2: Additional Information for Pedestrians	47
Figure 4.3: Virtual Connection through a park	49
Figure 4.4: Crosswalk Types.....	51
Figure 4.5: Corner Points Derivation Mechanism for Crosswalk Definition.....	51
Figure 4.6: Crosswalk Complex Example.....	52
Figure 4.7: Public Transport Information.....	55
Figure 5.1: Compass view for the iPhone Google Maps.....	62
Figure A.1: Use case model for the information system.....	76
Figure A.2: Services Use Case Model	78
Figure A.3: Routing & Navigating Use Case Model	81
Figure A.4: Search Use Case Model	87
Figure A.5: Configuration Use Case Model.....	90

LIST OF FIGURES

List of Tables

Table 4.1: Crosswalk Table Example	52
Table 4.2: Crosswalk Definition Table for Complex Example	53
Table 4.3: Crosswalk Definition Table for the BRIDGE type	53
Table 4.4: Crosswalk Definition Table for the TUNNEL type	53
Table A.1: Services Use Case Package Description	77
Table A.2: Routing & Navigating Use Case Package Description	77
Table A.3: Search Use Case Package Description	77
Table A.4: Configuration Use Case Package Description	77
Table A.5: Search Nearby Services Use Case.....	78
Table A.6: Select Service Use Case	79
Table A.7: Manage Favourite Services Use Case	79
Table A.8: Service's Determine User Position Use Case	80
Table A.9: Calculate Route Use Case	81
Table A.10: View Calculated Route Use Case.....	82
Table A.11: View Schedule for Public Transport Use Case	82
Table A.12: View Intersections Use Case.....	83
Table A.13: View Total Route Time Use Case.....	83
Table A.14: View Estimated Remaining Route Time Use Case.....	84
Table A.15: View Road Book Use Case	84
Table A.16: Calculate Alternative Routes Use Case.....	85
Table A.17: View Alternative Routes Use Case	85
Table A.18: Routing & Navigating's Determine User Position Use Case	86
Table A.19: View POI Information Use Case.....	86
Table A.20: Search POI Use Case	87
Table A.21: Search Address Use Case.....	88
Table A.22: Search Nearby Transports Use Case	88
Table A.23: Search's Determine User's Position Use Case	89
Table A.24: Search Transport Schedule Use Case.....	89
Table A.25: Configure Profile Use Case.....	90
Table A.26: Configure User Type Use Case.....	91
Table A.27: Configure Settings Use Case.....	91
Table A.28: Configure Routing Type Use Case.....	92
Table A.29: Configure Interface Display Use Case	92
Table A.30: Configure Services Use Case	93

LIST OF TABLES

Abbreviations

POI – Point Of Interest

HPF – High Pass Filter

CRUD – Create, Read, Update, Delete

GPS – Global Positioning System

GIS – Geographic Information System

GIS-T – Geographic Information System for Transportation

PND – Portable Navigation Device

ABBREVIATIONS

Chapter 1

Introduction

The dissertation "Information Systems for Public Transport Users" is presented in this document. The objective of this dissertation is the specification of a location-based solution for pedestrian users of navigation systems and public transport. With the evolution and proliferation of mobile devices, navigation became possible using these devices and pedestrian navigation was the next logical step. Pedestrian navigation is a topic that is still being explored and nowadays there are not many navigation systems which consider the needs of the pedestrian when navigating. Unlike vehicle navigation, pedestrian navigation presents new possibilities for navigation but presents some problems as well.

For the specification of an information system of this type, the needs and requirements of the pedestrian users as well as the problems that concern pedestrian navigation have to be identified possibilities as in how to solve them have to be presented as well. Higher degree of freedom, much more information regarding the public transport networks and pedestrian only navigable areas are just some examples. With the inclusion of services to enrich the user's experience and adaptivity to the user are also examples of possibilities which make these systems an interest and challenging area of study. In order to support all these functionalities there is a need to define a service-oriented architecture, a spatiotemporal data model that can contain all this information and since several means of transportation mean multimodality, a multimodal routing algorithm.

However, it should be still clear that, besides the advancements in technology, the development of such a system in mobile devices is still a hard task to do, mostly because of the limitations of such platforms. So, the architecture must be carefully thought in order to provide a robust and stable base for these systems in years to come.

1.1 Motivation

This dissertation has its origin in the need to implement a navigation system for public transport users. With the evolution of the capabilities of mobile navigation devices and the widespread introduction on the market of high performance smartphones, such as the iPhone 3GS or the HTC HD2, which possess geo-localization through integrated GPS, navigation applications with these devices are increasing. The fact that nowadays people are encouraged to use alternative means of transport like public transport, especially in big cities; the fact that people are becoming increasingly more attracted to the outside and the importance it has for a tourist not to get lost on an unknown city for example, make the need for pedestrian navigation systems more clear.

The fact that pedestrian navigation with the use of public transport information is still an emergent area with its potential barely exploited in the commercial market makes the specification of such a system important for all the pedestrian users whose navigational needs have not been attended with the current, almost inexistent pedestrian navigation solutions.

1.2 Problem Statement

In this dissertation, the main problem lies with the creation and specification of a solution for a multi-modal navigation system for public transport users. This solution should allow the user to navigate as a pedestrian or as a public transport user, to provide the user with feedback during the journey and to learn from the user in order to present him with a customized experience, whether it is in route planning, navigation aids or information presentation. The solution is expected to run on the device, only needing to connect to the internet to retrieve information from the services provided to the user such as real-time delay information on the public transport.

In order to do so, a service-oriented architecture has to be presented which considers all the requirements and functionalities that are expected of such a system, having in consideration the limitations of mobile devices.

Also, the data model has to be flexible so as to consider the various topologies correspondent to the various transport types, the schedules and tariffs of the transports while not being bound to a single city or country.

Finally, there is the need to present a multimodal routing algorithm that allows routing using the several layers of the graph topology correspondent to the several transport networks available and which uses the additional pedestrian and public transport information to present multimodal routes to the pedestrian user.

1.3 Objectives

The idea behind this dissertation is to present the specification of a solution based on location that considers the needs and requirements of a multimodal navigation system mainly aimed at the pedestrian users of public transport. The specification defines the useful services and functionalities the system should contain such as planning multimodal trips and providing schedule information for public transport.

Also, this dissertation requires a very well planned architecture in order to comply with the expectancies of the users, as it must support the functionalities presented in the specification of the solution such as aiding the public transport user in planning multimodal trips. Besides, this project is oriented towards mobile platforms such as PND and smartphones, and so the limitations of these devices must be taken into account in order to provide a robust and solid base for future pedestrian navigation applications. This architecture must be service-oriented and user-centered.

The definition of a spatiotemporal data model that considers and contains all information regarding pedestrian navigation only and public transport, such as their schedules and stop locations, is required as well.

Finally, the definition of a planning algorithm for the public transport user's trips is also required. The algorithm should be able to use the information previously mentioned in the data model in order to aid the pedestrian user in their route planning.

1.4 Expected Results

At the end of this dissertation it is expected that the resulting architecture is able to comply with the requirements proposed. Also, it is expected that a data model that allows the addition of all the expected pedestrian-only information as well as the public transport information is provided. To do so, the data model must be well documented in order for the developer of the maps to be able to add this information with the minimum effort possible and avoiding redundant information. Finally, it is expected that the outlines of the planning algorithm should be given as to how to perform multimodal routing along the map.

1.5 Structure of the Dissertation

This document is divided into six chapters. The present chapter 1 introduces the problem and describes what is expected of this work. In chapter 2 it is presented the main background information and state of the art. In chapter 3 it is presented and described the requirements specification for the problem as well as a possible architecture of the system. In chapter 4 it is presented the proposed data model and each of its components described. In chapter 5 it is described the advantages and disadvantages of having a sensor enhanced application and the details of the implementation are presented. Finally, in chapter 6 we present the main conclusions and final remarks of this dissertation and the future work to be done.

Chapter 2

State of the Art

2.1 Introduction

The advancements in technology, be it either in mobile phone, smart phones, PDA's or navigation devices, have made possible to take these information systems a little further. The needs of these systems' users have been increasing in the past few years both in number and complexity. Taking this increase in number into account, the need to study these new requirements gains a brighter spotlight in order to create new and improved navigation systems. Lately, with the growing encouragement for the use of alternative means of transportation (by alternative I mean other than the personal vehicle) and considering the new needs of the navigation systems' users, research turned to the pedestrian navigation systems and later to the use of public transport information in these systems. When it is said public transport information it is meant all information regarding these means of transportation like schedules, tariffs, stop locations, route-to-route intersection points (be it between the same transport type (bus to bus) or not (bus to train)).

As mentioned before, navigation systems have been evolving and pedestrian systems are no exception. In recent years, many proposals for improving multi-modal navigation systems have been published. These improvements vary from better determination of the exact position of the pedestrian to new approaches to data models or spatiotemporal data representation.

Investigation in pedestrian navigation systems for public transport users has been increasing in recent years but even with this increase there are still very few commercially sold solutions at the users disposal. This fact can be justified with the arguments that navigation systems for this type of user, the pedestrian, have different requirements and constraints and changes have to be made to the applications and with the argument of the great amount of information required for these types of navigation systems. Although both arguments can be refuted, the second one is not so easy to ignore because the data model is still one of the major problems for these systems. Unlike the traditional navigation systems (for vehicle navigation), the navigation systems for public transport users are not only systems with spatial data representation but spatiotemporal data representation because people have to keep up with the public transport' schedules.

In this section of the dissertation I will present the main existing products in the market for multi-modal navigation systems but mainly pedestrian navigation systems and those that make use of public transport information. I will also present the main topics and ideas of the

research in the academic literature, mainly focusing in the data model, data representation and trip planning algorithms.

2.2 Requirements for pedestrians

Navigation systems for pedestrians have to take into consideration a series of different constraints and requirements.

The main problems regarding pedestrian navigation are:

- the instructions for vehicles are confusing for pedestrians because for pedestrians it is more easy to take landmarks as references;
- the degree of freedom, i.e., pedestrians are fairly free in movement when comparing to cars which are restricted to the lanes in the street network;
- the velocity of movement of a pedestrian is a few times slower than that of a vehicle which makes is a problem for the accuracy of the determination of the pedestrians position and results in different perception of the surrounding environment;
- the spatial resolution is different since pedestrians can observe the environment in a more detailed way due to their reduced velocity.

Taking these problems into account it can be implied that the datasets in which vehicle navigation and pedestrian navigation are based cannot be the same.

When considering the public transport information, other problems also arise for the pedestrian navigation systems, namely the time dependency of the pedestrian and the time dependency of the public transport themselves. Public transport have to maintain schedules in order to function efficiently and the pedestrians that use these transports have to keep up with them to travel. This factor has to be taken into account as well in planning the trip.

All this information which is relative only to pedestrians and all information regarding the public transport has to be taken into account and included when creating the data model for the system. This data model needs to have information regarding:

- Stops for buses, subways and trains;
- Interchange stops for switching routes between the same type of transport or a different one;
- Routes for buses, which are parts of the graph considered for vehicles;
- Routes for subways and trains, which are unique for them (no other vehicle travels on them);
- Information regarding schedules and tariffs;
- Crossable areas for pedestrians only, like parks and squares.

But, aside from the problems that arise from the differences between pedestrian and vehicle navigation and considering the public transport information, the pedestrians have preferences over the services and information that can be provided by these systems.

According to Grotenhuis et al. [GRO07] and based on the thesis of Grotenhuis, customers of Integrated Multi-modal Travel Information systems (IMTI) for public transport have a series of requirements for these systems.

These travel requirements of the users are divided into three groups of information:

- **pre-trip information** which is the information regarding the trip planning step where the users prepare their future travel. In this step it is defined which tasks are to be carried out to reach the destination and in which order they are to be carried out;
- **wayside information**, refers to wayside locations like bus stops, stations, park and rides, among possible others. Important information to consider is the first stop and interchanges;
- **on-board information**, consists of information provided while on board of the transport and is always preceded by pre-trip and wayside information. This information is mainly used for travel support.

As it can be seen in Figure 2.1 presented below, in order to achieve a service with quality, two main dimensions of information have to be considered, time and effort saving information. Time can be saved by way of processing information and during the journey. Effort savings can be either physical, cognitive or affective.

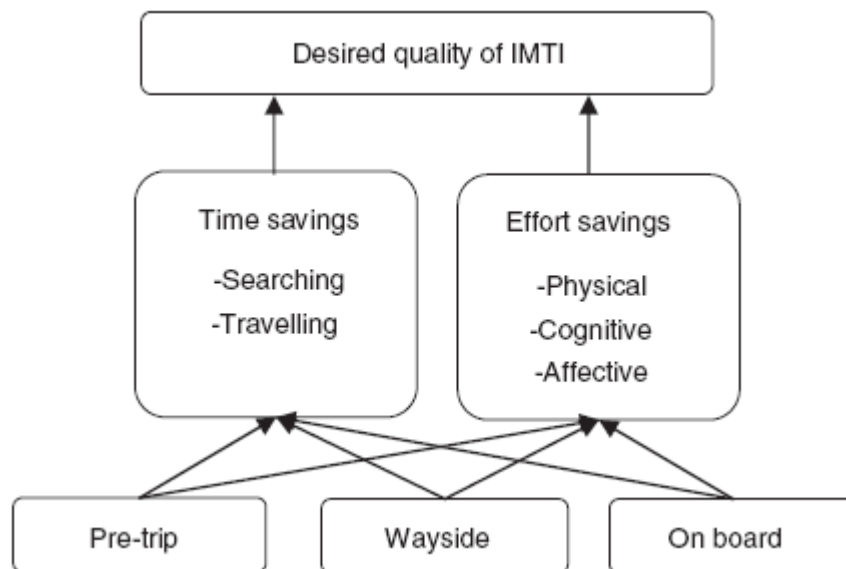


Figure 2.1: Source: [GRO07]

With the study conducted by the author these desired time and effort saving requirements it is possible to get a general view of which desires have to be considered on an IMTI system in order to produce and deliver a product with information quality.

State of the Art

The results obtained by the author are presented below for all three types of information: pre-trip, wayside and on-board information.

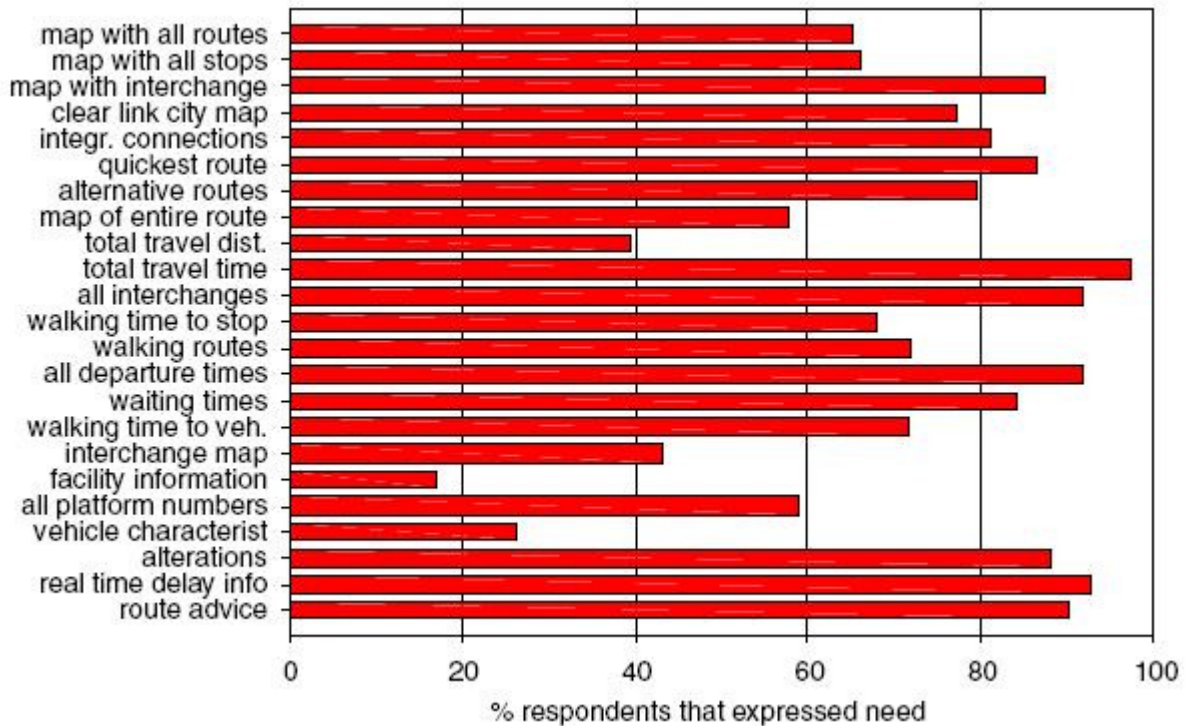


Figure 2.2: Source: [GRO07]

As it can be seen in the figure above, regarding pre-trip information, passengers have a greater need for mostly information to help save time. Mainly, passengers need information about real time delays, route advice to avoid delays and information that is usually given by route planners like the total travel time, all departure times and interchanges and the quickest route possible to take.

State of the Art

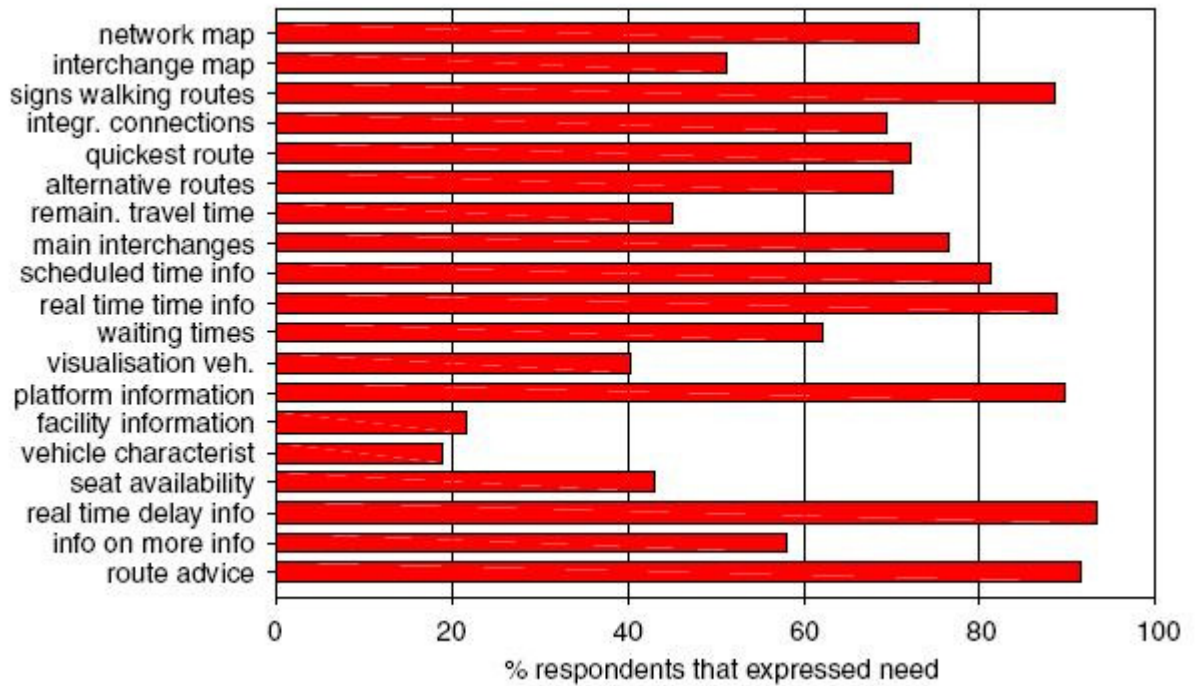


Figure 2.3: Source: [GRO07]

When considering wayside information, public transport passengers stated the need for real time information, mainly regarding quick connections when the passenger has to catch the vehicle on the right time. There is a lower demand on orientation, planning or information regarding the remaining trip and just like in the pre-trip stage, real time information on delays and route advice to avoid delays is the most desired information. Wayside information has a bigger emphasis on saving effort and travel time.

State of the Art

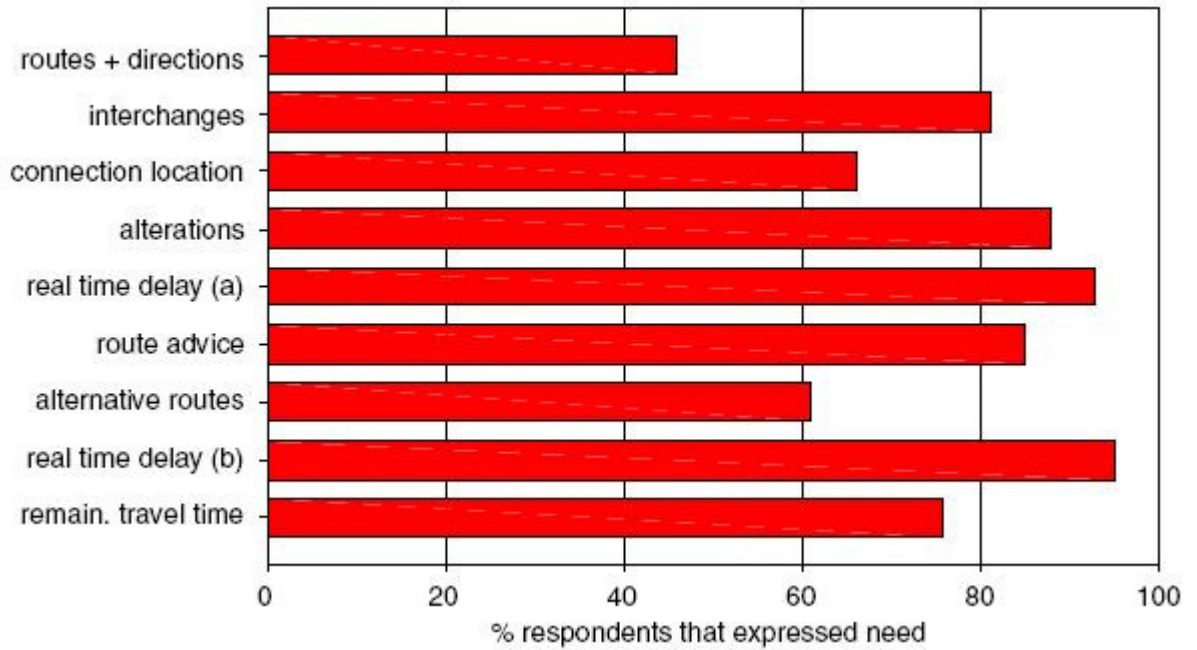


Figure 2.4: Source: [GRO07]

Finally, as it can be seen in the figure above, when considering on-board information, the main concerns of the passengers goes to the smooth progress of the remaining part of the journey which makes time-related information the most important along with real time information. Information about the route, the location of interchanges, crossing routes and directions are pointed out by frequent public transport users as of great importance to them in order to save mainly travel time and affective effort.

2.3 Problems in Pedestrian Navigation Systems

Pedestrian Navigation Systems have different requirements from the traditional vehicle navigation systems. They raise new possibilities due to its higher degree of freedom but bring along some problems which need to be solved in order to have a solid base for pedestrian navigation systems.

The main problems are listed and detailed below.

2.3.1 Speed

Unlike vehicles, pedestrians move at a much lower speed, typically 3 or 4 km/h. The low speed of the pedestrians influence the position determination by the GPS because it considers the different positions of the user as errors within the predefined threshold for the exact location of the user. The GPS only considers that the user actually moved only when his position suffers a change which is relatively big in distance, when considering the initial position and the last fix of the GPS (or actual position). There are several solutions proposed in the literature but some have advantages over the others because we are considering a pedestrian navigation system for a mobile device. Among these solutions there is the possibility of recurring to integrated sensors on the devices like accelerometers. The use of these sensors is an advantage because they are becoming a standard in recent devices like the iPhone 3G and 3GS, the HTC HD2, among others and they have been proven to work well, for example in the Google Maps application for the iPhone.

2.3.2 Position Determination

As mentioned before, the low speed of the pedestrians is one of the reasons why the determination of the position is harder for pedestrian navigation systems. However, this is not the only reason for this problem to occur. The traditional problems for vehicles like urban canyons also affect the position determination for pedestrians but the latest are affected greatly because of the afore mentioned low speed.



Figure 2.5: Urban Canyon

Once again, there are several solutions presented in the academic literature. Kalman Filtering [ROJ], Dead Reckoning and the Critical Point Method and Location-aware State Machine by [BAR08] are possible solutions to enhance the GPS signal determination and therefore the position of the pedestrian user.

2.3.3 Navigation aids

Because of the low speed at which pedestrians travel their spatial recognition is higher. Pedestrians observe their surrounding environment with a higher level of detail and that calls for the need of different navigation aids, both audio and video. Pedestrians have an easier time navigating in unknown environments if they can guide themselves through visible landmarks instead of actual directions, like what is used in vehicle navigation. Also, in these unfamiliar environments, without these visual landmarks it is possible and normal to lose the sense of direction in which the pedestrian is facing, especially if the roads are very similar. A possible solution for these cases is to recur to integrated sensors in the devices, like magnetometers which act as digital compasses and make it possible to know the true and/or magnetic heading of the user. These sensors are starting to appear in mobile devices and an example of one is the iPhone 3GS or the Google Nexus One. Another solution is the use of augmented reality to display visual information about landmarks which are relatively near.

2.3.4 Information

One of the main problems concerning pedestrian navigation systems is the amount of information available and how to represent it and deliver all this new information to the user. There is a lot of information only for the pedestrian to consider when navigating. There are sidewalks, crosswalks, pedestrian only crossable areas like parks and squares. There may be stairs or ramps along the referred sidewalks and crossable areas. There is also information regarding the various public transport systems like bus routes, train routes, subway routes, stops and schedules for all of these routes and maybe even information about public transport zones

and tariff costs. The solution for this problem comes from the definition of a custom Data Model that takes into account all this information.

2.3.5 Adaptivity of the system to the user

One of the most desired features of a pedestrian navigation system is that the system adapts to the user by learning from his actions and according to a group of configurable options, which make the user's profile. With the inclusion of new types of users there are several new concerns that arise for several parts of the system, mainly at the routing level and the heuristics used and the way information is given to the user for navigation purposes. In order to obtain such adaptivity most modern systems use either ontologies and semantic knowledge or have a rule-based system architecture type.

2.3.6 Routing

Because of the higher degree of freedom of the pedestrians when compared to vehicles and all the new information for these users the routing algorithm cannot be the same. Routing of the pedestrian has to be calculated differently. As mentioned before, the inclusion of information about the public transport systems means several types of public transport, like subways, buses and trains. This means multimodality. Multimodality brings problems because it makes routing more complex. The bus network of a city is not very different from a subset of the transit network that lies underneath it, but in case of subways and trains, they need an almost completely separate network, only connected by the POIs that represent stations. Another factor that contributes to increase the complexity of the pedestrian's routing is the time component of the public transport. The information regarding public transport' schedules transforms the spatial data model into a spatiotemporal data model and as such, the routing algorithm has to cope and use this time information in order to provide a usable shortest route to the pedestrian user. Finally, if the routing algorithm is going to be multimodal and multi-objective there is a need to have proper heuristics to consider the information of the profiled user and use accordingly to this profile. For example, if the user is handicapped it makes sense that the route provided to the user will be one where the route planner will try to avoid stairs as much as possible.

2.4 Information systems for public transport users

2.4.1 Existing solutions

As mentioned before the evolution of technology regarding navigation and mobile devices made possible to create these new types of navigation systems which are more targeted to public transport users. Despite this evolution in navigation systems, there are not many solutions still. There are a few which are already commercially sold and new ones are appearing but their number still is scarce. The main solutions in the market will be presented as well as their capability and features included.

One of the solutions in the market is the Navitime from Japanese company Navitime Japan Co., Ltd. This product is available in both PC and mobile application versions but for the purpose of this dissertation only the mobile application will be presented.

This product integrates public transportation methods operating on a time table, self-driving and taxis and navigation by foot, allowing users to plan travel multi-modal routes. Included in these multi-modal transportation types are included walking, driving and public transportation methods such as buses, trains, subways and taxis. As for navigation aids, Navitime includes voice and text guidance for both pedestrian and car navigation. The application is available for a variety of mobile devices with different operating systems such as Windows Mobile 5/6 Professional and 6 Standard, Symbian S60 3rd and 5th Editions, BlackBerry or Java MIDlet version for other operating systems.

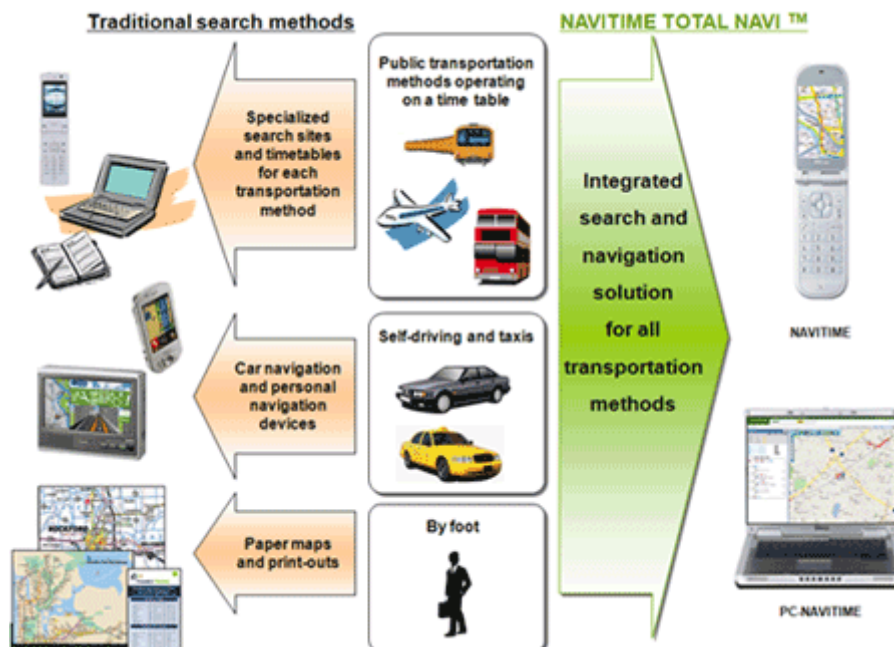


Figure 2.6: Navitime Supported Transport Types

Among its features is the possibility to compare and choose among the travel route calculated from a combination of walking, driving, riding a public transport and flying. It includes real time traffic information for car navigation as well depending if it is available at the current location and it requires an Internet connection for accessing this service. It provides the user with the cost and time of travel whereas the time tables are available for public transport in that area in order to calculate the actual time. As extra features it provides the user with the CO2 emissions data so as to allow the user to choose more environment friendly routes and the estimated calories burned when walking are provided as well. This solution is already available in a number of countries in Europe, in the United States, Australia, Japan and Singapore/Malaysia.



Figure 2.7: Navitime Walking Route Example

It can be seen in figure 6 that in some cities walking information for pedestrians only, like sidewalks, are already considered giving the user a more real experience.

This application can run in a wide range of mobile devices since it features a light map drawing engine and a light distribution format, which is needed due these devices small memory.

Google recently launched the walking directions feature as an increment to the car and public transport navigation previously available in Google Maps. This feature still has a few rough edges since not all information regarding pedestrians is available. New information like pedestrian bridges or the streets which have sidewalks is being collected in order to provide the user with a more pleasant and efficient experience.

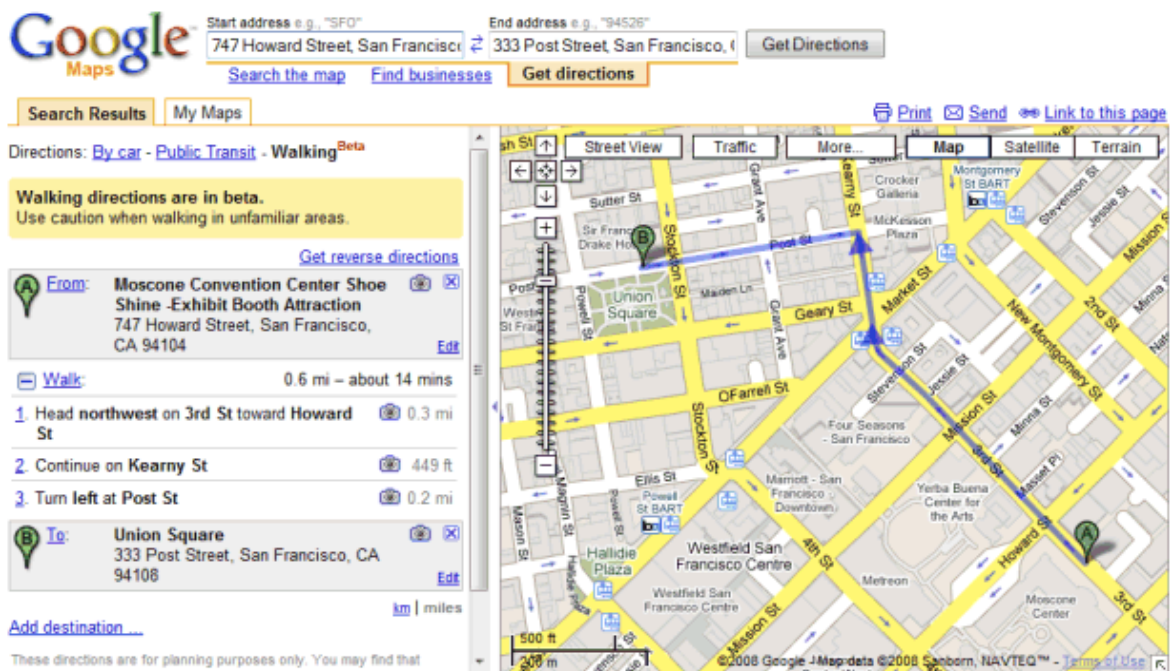


Figure 2.8: Google Maps Web Interface now with walking directions

2.4.2 Planning algorithm

The planning algorithm is one of the key points of a navigation system. It can determine whether you get to your destination in a determined time window by giving you the fastest possible route if your main concern is time, or if you are on vacation get you to make the most of your time by giving you the optimal route for sightseeing, for example. It is one of the main research topics as well because of the many possibilities given the number of algorithms available, specifically for planning purposes or not. The presented planning algorithms in this section are divided by approach.

2.4.2.1 Headway-based model and Schedule-based model

When dealing with trip planning in transit networks, several problems arise when comparing them to highway routing and traffic assignment problems. Transit planning problems derive from their highway counterpart because they add more complexity to these problems and because the very network has significantly different characteristics from highway networks [SPE94]. As referred by Peng et al. [PEN00], the main problems with transit networks are that, first, they are time dependent networks, different times of day and days of the week have different levels of transit service. Second, one street segment may have several bus routes and stops and many bus routes may stop at the same bus stop, also referred to as the "common bus lines problem" [CHR75]. Third, in transit networks the origin/destination pair may not be symmetrical with that from destination to origin. Fourth, transit transfers are dependent from the time of arrival of the other bus, for example, and the best path between two points may change

depending on the time of day or day of week. Also, the turn weight (or wait time) is more difficult to determine because of the fact that one street segment may have several bus routes and that each one has its own headway. With the additional complexity added by the fact that some buses stop serving at certain times of the day, these problems can be approached as time dependent constrained shortest path or TCSP.

Because of the complexity of the transit route network itself and according to Peng et al. [PEN00] most of these prior models for shortest path finding could be divided into two different groups, headway-based models or schedule-based models.

The headway-based path finding algorithms assign passengers to the first transport that arrives based in the combined frequencies of common bus lines [SPI89] and assumes a constant headway on a segment during a time period.

The Schedule-based transit network models determine the time dependent least cost paths between all origin/destination pairs in the transit network using a branch-and-bound type algorithm. They assume that the passengers board the first vehicle to arrive according to the previously determined schedule. Despite the fact that the headway based is stochastic and heuristic in nature, its algorithm itself does not identify which routes the passenger should take if there is more than one possible path or bus line going to the same destination. On the opposite, the first vehicle to arrive according to a fixed schedule is deterministically identified in the schedule-based transit models. "Therefore, it yields one and only one optimal solution for any given origin/destination pair" [PEN00].

As referred, one of the main problems with the planning algorithm for a navigation system for public transport users is the fact that the whole system will function on a time dependency. Public transport are time dependant because of their schedules and the users have to depend on time while planning a journey. There is a need to take into account the walking time from the current position or from another origin point to the closest public transport stop, the travelling time while riding the public transport, the time to switch itinerary and possibly the waiting time for the other public transport to arrive and an extra possible walking time to destination, after leaving the transport, for example.

2.4.2.2 Combining the Headway-based model and Schedule-based model

A study by Peng et al. [PEN00] showed that the combination of these two models in a two stage path finding process can be used to fill in the gaps caused by the increase in complexity of the transit network. In the first stage, the average headway is calculated for each segment of the transit network and then used to calculate travel time on each link and the turn weight at each intersection. The average headway must be calculated for each time period since transit service frequency varies along the day. In the second stage, in order to estimate the shortest path between the origin/destination pair, a vine-building type shortest path algorithm is used. This shortest path problem is transformed and treated as many time dependent shortest path problems over a discrete time interval.

This hybrid method may not produce the shortest path since it relies on the average headway as turn weight. It has to calculate the actual time at each potential transfer point based on the service schedule in order to produce one single best route.

This algorithm fits the unique characteristics of transit networks like its time-dependent services, multiple transit routes on the same street and the non-symmetry of shortest paths between the origin/destination and destination/origin pairs.

2.4.2.3 Use of Semantic Knowledge

A different approach was proposed by Chen et al. [CHE08] that resorts to the use of semantic to allow a better determination of the best path for the trip. Instead of using the path finding algorithm right away, in most cases the shortest path algorithm, it uses ontologies to group user preferences and requests making service discovery more efficient and combines lesser services into an integrated service for satisfying a request. In order to turn the multi-constrain path selection problem associated with transit networks, it aggregates available services into semantic clusters according to their attributes and semantic information. Service composition is then executed and first, one shortest path is determined among these clusters and then in each cluster a suitable service is chosen, forming the complete path.

The path finding algorithm is a three step process. In the first step, the semantic knowledge is aggregated by the aggregator and saved in a database. In the second step, the composer determines which is the best path for the origin/destination pair requested by the user by determining the best shortest path among the different semantic clusters and then the best service in each one. Having the complete path formed, the system presents the composed path to the user.

The system uses a graph (G) divided into two layers, a syntactic and a semantic layer. For the syntactic layer, S is a set of available transport services and TR is a set of available transport services. Each tr_x in TR indicates a service segment from a start station (si) to an end station (sj). As for the semantic layer, L represents a n-levelled tree of places where an internal node (non-leaf) li in L represents a place which in turn can have children nodes representing more detailed divisions. Each of these nodes can have as well a parent node that represents a broader place covering li . All stations map to one and only one leaf node in L.

TS is a set of semantic transportation services where each ts_x indicates a link between two places (li, lj). These two places must have the same parent node and there exists a non-empty subset $TR' \subseteq TR$ such that tr_x in TR starts from a station mapped to li or to one of its descendants, and ends at a station mapped to lj or one of its descendants.

In order to aggregate the services into the semantic clusters, the system considers as input:

- a graph of the services (G) with an empty L and TS;
- a threshold for aggregating stations (*PedThrd*) that indicates the maximum distance that stations can reach to each other without transportation services;
- a threshold for aggregating places (*SrvThrdOffset*) that indicates the incremental distance offset in each semantic level.

As an output, the system returns the input graph G with the semantic transportation service TS and L .

Quoting the authors, the aggregation process follows three main phases:"

1. Aggregate the stations in S such that each station maps to one and only one leaf node in L . As shown in Fig. 3a, $\{s_1, s_2\}$, $\{s_3, s_4\}$, and $\{s_5\}$ are aggregated and mapped to the formed leaf nodes l_1 , l_2 , and l_3 in L .
2. Aggregate the transportation services whose distance lie in the range $[base, base+SrvThrdOffset2]$ in TR . As shown in Fig. 3b, tr_1 and tr_2 are in the range, thus tr_1 and tr_2 map to ts_1 and ts_2 respectively.
3. Aggregate the places in L , such that the places have a parent if they are connected by semantic service in TS . As shown in Fig. 3c, l_1 and l_2 is connected to each other by ts_1 and ts_2 , such that l_1 and l_2 have the same parent l_4 .
4. Increase the restrict range and repeat phases 2 and 3 until the distance base is greater than the maximal service distance of TR such that all places and services were aggregated.

Note that step 1 is a preliminary step to the main phases referred by the authors which are steps 2, 3 and 4. The result is shown in figure 8d.

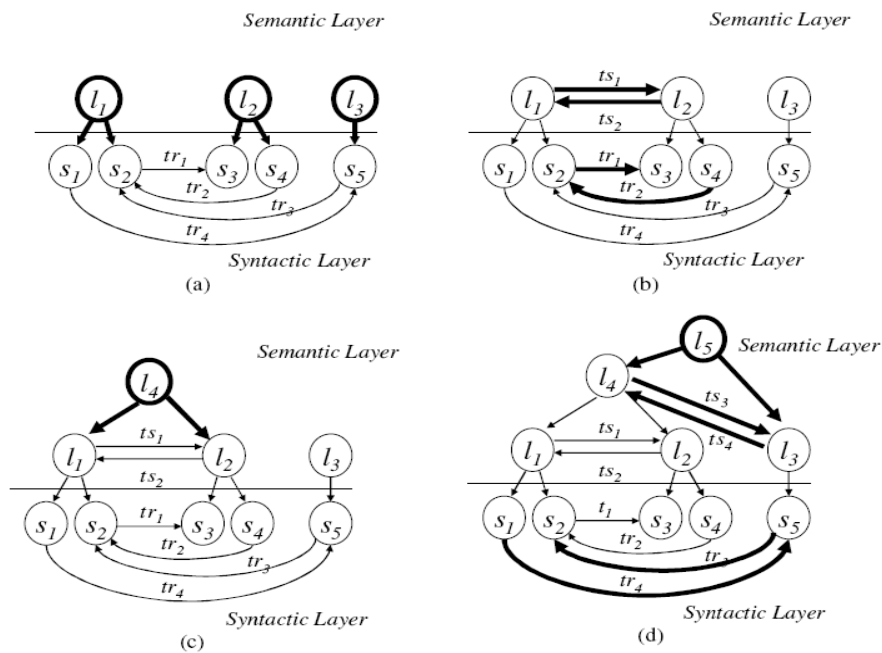


Figure 2.9: NimbleTransit's aggregation process. Source: [CHE08]

In the end of the aggregation process, a semantic layer hierarchy is generated in L and used by the composer to construct transit paths. The composition process takes as input:

- s_s , the source station;
- s_d , the destination station.

The result or output of the composition process is a transit path $PSE = \{pse_1, pse_2, \dots, pse_n\}$ which is a sequence of services where all $pse_i \in PSE$ belong to TS or to TR and for each $pse_i \in PSE$ the end of pse_i is equal to the start of pse_{i+1} ($EndOf(pse_i) = StartOf(pse_{i+1})$).

The composition process follows three main phases where the first one is a preliminary step, just like in the aggregation process. Quoting the authors, "

1. Find the common ancestor of ls and ld which are mapped to ss and sd respectively. If there exists a common ancestor, it means that the two stations can reach each other.
2. Find the semantic transit path $tsPS$ from the semantic place lsa to lda . lsa is the ancestor of ls and lda is the ancestor of ld ; both lsa and lda are the children of their common ancestor.
3. Select the syntactic transit path $trPS$ through $tsPS$.
4. Compose the rest transit path by recursion. If the selected path $trPS$ is $\{tr_1, tr_2 \dots tr_n\}$, then the process keeps constructing path from ss to $StartOf(tr_1)$, $EndOf(tr_1)$ to $StartOf(tr_2)$, ..., and $EndOf(tr_n)$ to sd ."

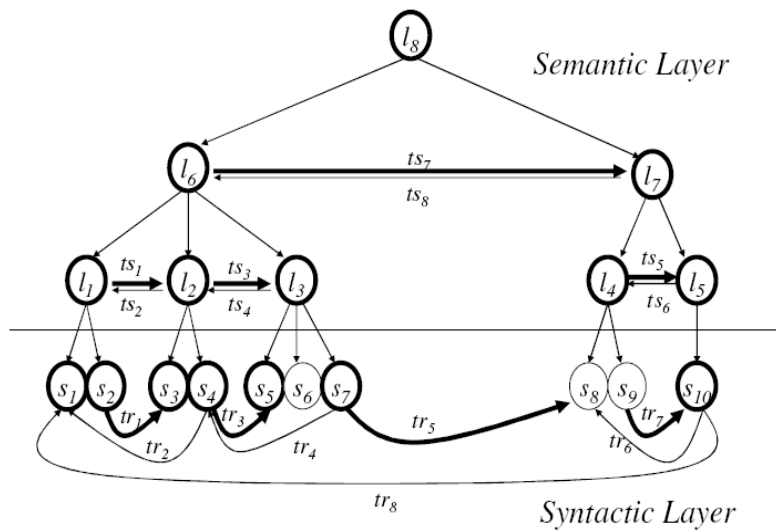


Figure 2.10: Example of result from NimbleTransit composition process. Source: [CHE08]

The result can be seen in Figure 9 where the source station is s_1 and the destination station is s_{10} .

Still, the use of the shortest path algorithm creates a bottleneck in the composition performance.

The approach taken by the algorithm of this system has good results due to the fact that, with the help of the semantic knowledge, it can limit the number of stations to search. By reducing the number of points, complexity is decreased and a faster result is given to the user. Nonetheless, the fact that this approach is run on a PC with a 2.40 GHz processor and 2GB of memory can make it not so fast on a mobile device. The idea itself could be good to help in the search for a good path planning algorithm with the downside that it relies heavily on an Internet connection since it is Web Service based.

2.4.2.4 Multi-objective optimum path algorithm

This algorithm by Ziliaskopoulos et al. [ZIL07] determines the optimum path based on priorities and target values of duration and cost of the trip defined by the user's request. The user defines the origin and destination points, the terminals graph is built according to the information of terminals, links, schedules and costs and then the closest terminals to the origin/destination pair is determined. It then determines the feasible paths that serve the user's time and cost requirements. This algorithm is relevant to this project because it provides a base for a multi-objective optimum path algorithm.

2.4.2.5 Multi-modal, multi-criteria path planning with time windows

Considering the time dependency of the transports to their schedules and the dependency of the user to the schedules of the transports as well, it is then feasible to create an algorithm which takes these matters into account and calculates not only the travelling time but also calculates and gives the user a time window to leave in order to get to the planned destination in the allotted and previously determined time.

This algorithm developed by [ZOG08] is the solution these authors found for a multi-criteria shortest path problem with time windows in a multi-modal time-schedule network with time dependent travel times. These windows of time that are considered in the problem are much alike the earliest start and latest finish time values that can be found in the Critical Path Method (CPM) and they represent, in this context, the earliest start possible to start each of the steps of the journey, like walking to a bus stop or changing between bus A to bus B; the second value, the latest finish, represents the latest time possible to finish each of the steps of the journey in order to arrive the destination in the desired time window.

The solution presented by these two authors for the itinerary planning problem falls into the schedule-based approaches that were explained earlier in this chapter. It enhances the existing approaches by:

- incorporating the multiple criteria through lexicographical ordering;
- allowing the departure and arrival times to lie within the specified time windows;
- assuming time dependent travel times;
- taking into account the special case where an intermediate stop at a given time window should be scheduled.

2.4.2.6 Pattern First Search

Pattern First Search is an algorithm developed by Huang [HUA07] and used in the Object-oriented approach by Huang et al. [HUA08]. First, Patterns are "the geographic paths over which trips travel" [HUA08]. The Pattern First Search algorithm searches the network by pattern objects and since the topology of the system is defined uniquely by sequences and collections of objects, like transfer nodes and stops, the schedule-based Pattern First Search algorithm's use is facilitated. This organisation of transfer nodes by linear pattern objects reduces complexity of computational time and therefore providing a way for this, despite alternative, algorithm to present good results. It suffers from the negative aspect that despite presenting good ideas, it needs the systems topology to defined in objects and not by the traditional edges and nodes.

2.4.3 Data model

The data model is a key component of every navigation system. In it are contained the sub-components which make the navigation system function. It is responsible for defining the information contained in the system and how it can be represented. A data model for a public transport system as to contain information about the transit, bus, subway and train networks as well as the information regarding the planning algorithm and navigation aids. There are two main approaches for representing the topology of the networks, Edge-based, which is the most usual, and Object-oriented, which started to appear in the literature not so long ago.

2.4.3.1 Object-Oriented Vs. Edge-based (Vector)

An alternative approach is presented by Huang et al. [HUA08] which uses an Object-Oriented spatiotemporal data model against the traditional Edge-based (or Vector-based). In this approach the dynamic topology, components and services of the system, which change over time, are modelled as spatiotemporal objects that have properties and behaviours. Since the system is dynamic and highly time dependent because public transport work according to schedules, a spatiotemporal data model for dynamic transit networks allows the dynamic creation of an active network topology which represents the real-time network state and reduces topological redundancy, hence reducing computational complexity. This reduction in complexity enhances path-finding performance because the complexity in the schedule-based path-finding algorithm is reduced as well.

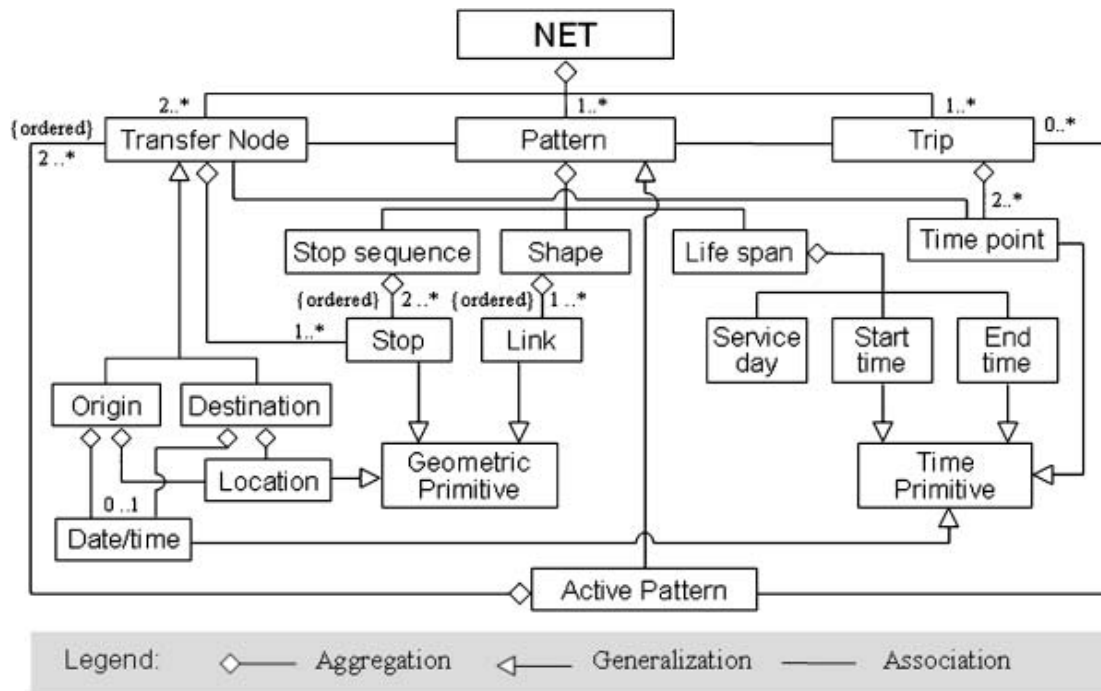


Figure 2.11: Class diagram of the transit network. Source: [HUA08]

2.4.3.2 Use of Semantic knowledge

On recent years, many researchers have been investigating how semantic knowledge and ontologies could be used in Geographical Information Systems. Zipf et al. [ZIP06] proposed a model in which ontologies were used to create a model for adaptive mobile Geographical Information (GI) services. This model allows the representation of personalized tour with real GI services like spatial queries or tour planning that adapt to the context in contrast with the Web-based or mobile systems that adapt textual information according to a range of user and context factors.

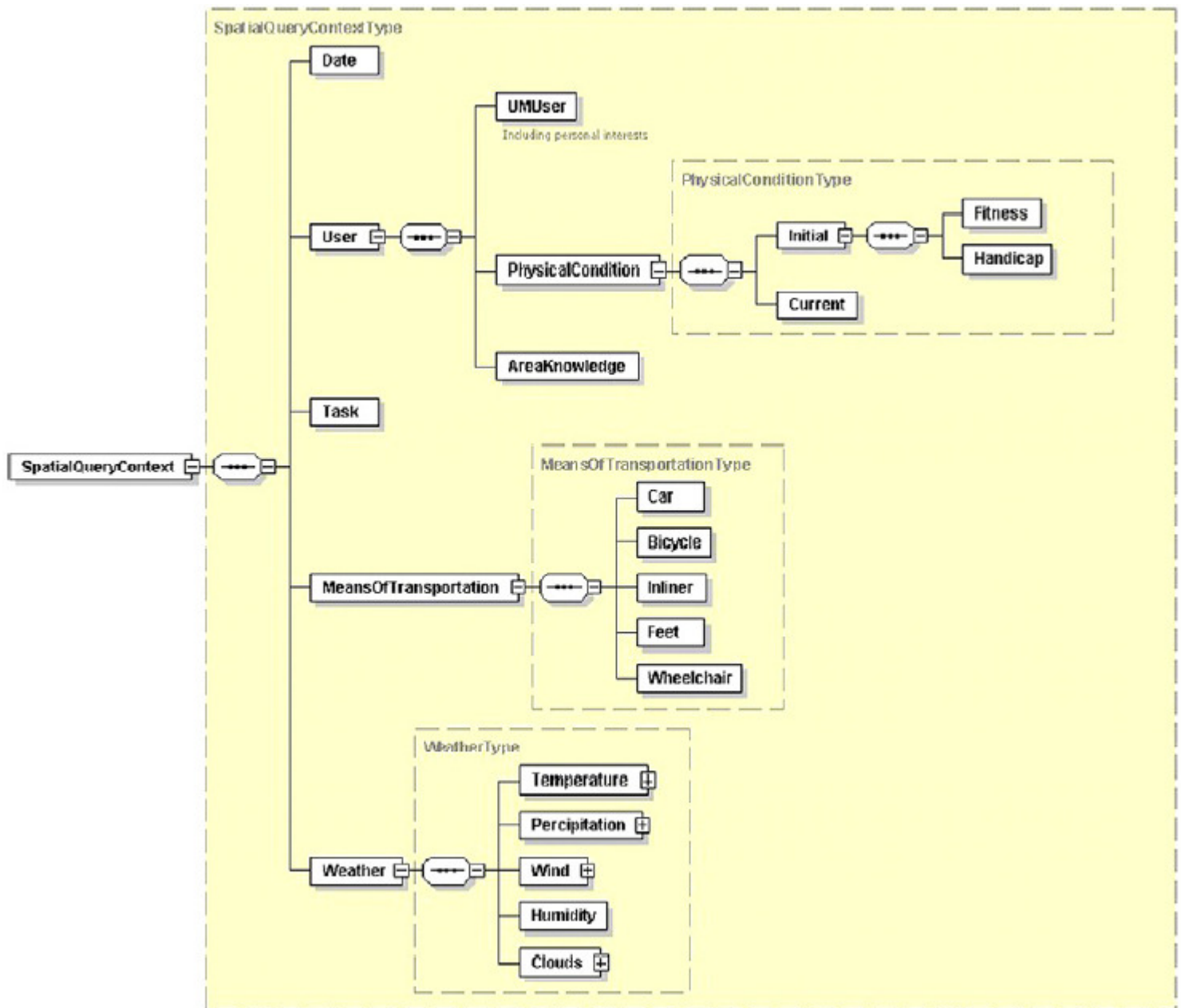


Figure 2.12: Context and user aware spatial query schema. Source: [ZIP06]

2.4.3.3 Category Theory

In 2007, Frank [FRA07] suggested the use of category theory as a means to complement the needs of public transport users that cannot be satisfied by the navigation systems. For example, a public transport user needs to go from A to B but needs to buy a ticket for that train which satisfies the trip. Existing navigation systems can give the user the activities to perform to go from A to B but they cannot tell the user which actions to perform, or where to perform them, in order to buy the ticket for the transport. He suggests the use of action - state - decision - action schemas associated with the navigation system and incorporated in the results given by the trip planner of the system. These schemas function much alike the critical path in large projects (CPM). Category Theory helps in creating a combined state-transition function between the two wayfinding tasks (navigation in space and "navigation in business logic" [PON06]). Despite showing some problems it could contribute by adding activities regarding public transport which could help achieving a more realistic experience to the user and freeing him from legal problems.

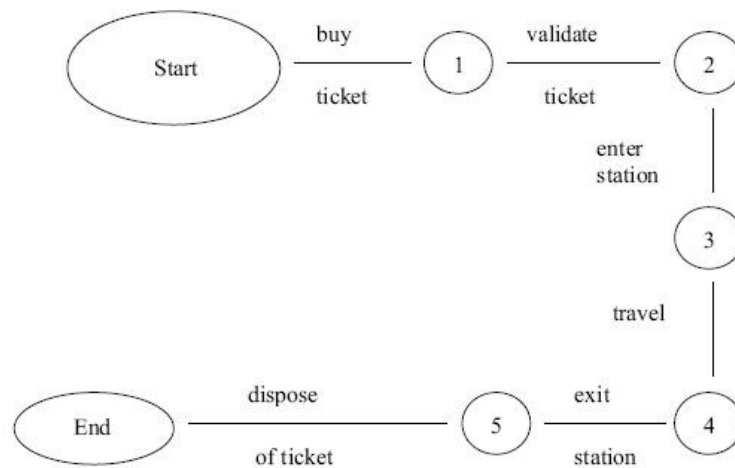


Figure 2.13: Graph describing actions necessary for a legal ride on a metro. Source: [FRA07]

2.5 Chapter Summary

Navigation Systems for Public Transport Users are complex. The amount of information that has to be considered is high and together with the time dependent services, it contributes to high complexity. In order to be appealing, the system has to be defined and created according to the needs and requests of its users. It has to learn from the user's behaviour and adapt accordingly in order to provide the right information at the right time. Also it is important that time, effort and cost are saved as much as possible to the user.

There are 2 main components in these systems, the planning algorithm and the data model.

Planning Algorithm

There is a lot of research on planning algorithms but mainly, the ones used in these systems are variants of the A* or the shortest path algorithm, with predominance to the latest. A simple shortest path algorithm cannot be applied in these cases if good results are to be expected. The time component of the public transport has to be taken into account. So, these shortest path algorithms fall into two main groups, headway-based and schedule-based.

The approaches presented show that investigation has been progressing to a multi-objective path planning algorithm for multi-modal navigation systems. This evolution is important because the information provided by the pre-trip planning is used throughout the trip. The main problem is that in order to plan a multi-objective trip, the planning algorithm has to decompose the trip plan into smaller problems in order to deal with the time complexity of the schedules and the multi-modality of the transports available. Also, the complexity of some of the algorithms available tends to make them hard to execute in a mobile device, making it only possible to be available as a service.

The Pattern First Search algorithm, despite having good results due to the reduced complexity provided by the data model, depends on having an Object-oriented model with pattern objects to achieve these results. Therefore, some notes can be taken in order to try and adapt them to other path planning algorithms.

Data Model

The data model of a public transport navigation system has a great deal of information to be modeled. This tends to be a problem, considering the limited storage capacity for a mobile device.

Behaviours that public transport users need to have during the trip (the "business navigation") is important to provide a more complete experience to the user, especially if he is not familiar with the public transport of the city. However, problems can arise from the inclusion of such behaviour schemas which can create redundancy in the system, like different business navigation on different cities or even stations within the same city. There has been an increase in the investigation of Object-oriented models for the topology of the network instead

the traditional Edge-based and despite having good results, companies tend to follow the traditional modelling way, at least for now.

Ontologies and semantic knowledge

Ontologies and semantic knowledge seem to be in the process of researching where and how to apply them in both the planning algorithm and the data model. Some of the ideas found in the literature are interesting, especially when it comes to classification of information and context awareness for providing adaptive geographic information services to customize the user's experiences according to his/her needs.

Chapter 3

A Location-Based Solution

3.1 Introduction

An information system for public transport users is a complex system. Such a system has a lot of functionalities and information to cover thus making the definition process a long one. In order to start the definition of such a system an adequate methodology is important. After a good methodology is chosen, the requirements specification for the system is defined in order to have a basis for the definition of the architecture and what functionalities and characteristics the architecture should possess in order to comply with the specified requirements. Finally, after the requirements specification, the architecture is then defined, presented and described.

So, in this chapter it is presented the chosen methodology, the requirements specification and a possible architecture that an information system for public transport users could have.

3.2 Methodology

In order to comply with the objectives, this dissertation was guided by an useful philosophy. The first step was to write down in a list all the basic features that were required for this location-based solution. After listing these required features, it was necessary to define an architecture that could comply with the requirements and the data model. In first place were defined the basic parts of the architecture or the data model and then the requirements would be checked to see if the requirements were being fulfilled and then, move to add a more complex feature, always in this cyclic way.

The ideal pedestrian navigation system is a complex system which has a lot of functionalities to support the user (the pedestrian) and to increase its usability. So, in order to support this vast number of functionalities, a system like this needs to have a solid, robust and well designed base, its architecture. Modularity, low coupling and usability are words of choice when thinking on all the parts that need to be assembled to create this pedestrian navigation system.

The ideal pedestrian navigation system would be one that would:

- support multimodal navigation and routing, both indoor and outdoor;
- have detailed information about the transport networks of the various transport types such as schedules and costs;
- provide the user with various services that would enhance his experience, such as real-time delay of the transports [DZI06] or calling a cab to his location without the need to make a phone call;
- allow the user to know his orientation at all times;
- support augmented reality for better orientation and navigation;
- give the user information for pedestrian only navigation like the pedestrian crossable areas such as parks and squares, sidewalks and stairs.

3.3 Requirements Specification

It is essential to establish correct requirements and specifications early in the development process to prevent errors later on in the system life cycle. Most projects include a body of information that describes the product or the project deliverables which deals with the objectives of the final product, defined in the project requirements (or software requirement specifications), and any rules for creating the product, defined in the project specifications.

A key benefit of developing a software requirement specification is in streamlining the development process. The developer working from the software requirement specification has, ideally, all of their questions answered about the application and can start to develop. Since this is a functional specification that was client approved, they are building nothing less than what the customer wants. There should be nothing left to guess or interpret when the functional spec is completed. This is the brilliance of the software requirement specification.

In this section it is presented the requirements specification in the form of user stories which is a software system requirement formulated as one or more sentences in the everyday or business language of the user. For more detail on the functional requirements, please consult appendix A.

User Story 1 - Off-line e on-line functioning

As a user, at any moment it should be possible to use the application on the mobile device in off-line mode to perform certain operations. The on-line mode should be used only when absolutely necessary.

User Story 2 – User type

It should be possible for the user to configure a number of options that enable the system to classify the user in a particular type, which will influence travel planning, such as transport type, or which route best suited to this.

User Story 3 - Multi-objective route planning

It should be possible for the user of the mobile application to plan a trip with several objectives and for that purpose specify a starting point and a destination and other goals he wants to see included in the planning. The planning is done taking into account the type of user, and should consider all means of public transport and also supported vehicular and pedestrian navigation (own vehicle).

User Story 4 – View planned route

It should be possible for the user to see the planned route.

User Story 5 – View estimated remaining trip time

The user, after defining a trip should be able to see the estimated remaining time to the end of the trip.

User Story 6 – View waiting time for public transport

The user should be able to see / be informed about the time that he has to wait in a particular stop of a transport.

User Story 7 – View departure time or frequency of public transport

The system should enable the user to consult the departure times of a particular public transport (or at least the frequency).

User Story 8 – Route adaptivity

During the execution of a trip and after the user leaves the calculated route the system should provide the user with a new alternate route.

User Story 9 – Interchange warnings

The system should provide the user with relative advance on the trading of faster vehicles to ensure that the user is forwarded in time to continue the route that allows you to reach your destination within the specified time.

User Story 10 – View route estimated total time

After defining a particular trip the user should be able to see what time is needed to perform the trip (depends on the means of transportation chosen).

User Story 11 – View alternative routes for a planned route

The user may choose alternatives to the planned trip when they exist.

User Story 12 – Navigate using planned route

It should be possible for the user to choose whether he wants the system to give him visual or audio-visual information to support him in the execution of planned trip.

User Story 13 – Augmented reality

As a user it should be possible in some mobile devices to aid the user in the execution of the planned trip using augmented reality.

User Story 14 – Information update

The user should be able to check whether there are changes to the data he has and update the data if desired.

User Story 15 – Update solicitation

The system should send a warning to the user requesting an update of the information they have on the mobile device, it is checked that there are changes to information.

User Story 16 – New information and contents

The user should be able to enable the operation anytime online to ordering information on a new city / country to be able to travel planning between new points of which you do not have information. To this should have the online mode is active.

User Story 17 – View public transport schedules

The user should be able to consult the schedule of a line of a given transport.

User Story 18 – View costs for public transport

The user should be able to consult the public transport fare for a particular trip on a particular transport type.

User Story 19 – View planned trip cost

The user should be able see the total cost of the trip, taking into account the public transport network that will travel on the trip and calculated in accordance with existing tariffs of that transport.

User Story 20 – Real-time services

It should be possible for the user to connect and receive information from services in real time, including delays on public transportation.

User Story 21 – Service Management

The user should be able to see the services available and manage them by configuring the system's service settings.

User Story 22 - Orientation

The user should be able to use the magnetometer sensor to be able to know what his current orientation at any time is, whether or not inside buildings and Urban Canyons, in order to enhance the user's experience. It can also be used in conjunction with GPS guidance, when available, to further enhance this experience.

User Story 23 – Pedestrian movement detection

It should be possible to use the user accelerometer sensor to detect movement by the pedestrian, and thus know more accurately when he is moving or not.

User Story 24 - Indoor Navigation

The user should have the ability to navigate within buildings whose mapping is available.

User Story 25 – View POI information

The user should be able to view information regarding a particular searched POI and its location on the map.

User Story 26 – View address in map

The user should be able to view a searched address in the map.

User Story 27 – Search for information

The user should be able to search for information such as POIs, addresses and zip-codes.

User Story 28 – Configure interface display

The user should be able to alter the definitions of the interface display in order to configure the information displayed and the look and feel of the application.

User Story 29 – Configure Routing type

The user should be able to configure the routing type in order to better suit the user type by doing so altering the route calculation process.

User Story 30 – Configure Settings

The user should be able to configure the application settings in order to personalize the application to his taste, for example, by changing the metric system.

User Story 31 – Configure Profile

The user should be able to configure the settings for the definition of a profile and therefore adapt the application to suit him better.

3.4 System Architecture

In order to support all the desired functionalities the system needs a solid and robust foundation, its architecture. In this chapter we describe such an architecture that would allow the system to support these functionalities. The typical architecture for this type of system is composed of several architectural patterns.

Starting by the beginning, the first big problem is to achieve portability between several platforms. This could be implemented by modifying the system's core code base for every platform. However, this is not a desirable solution mostly due to two facts: in the first place, no one can know in beforehand how many changes will be needed, in order to compile and run in another platform; but most importantly, no one can predict what will be the impact of those changes for the rest of the system's classes that are directly or indirectly dependant of these base classes.

Due to these facts, the first architectural decision was the sub-division of the system in four main layers [GAR03]. The main architectural pattern that composes this architecture is the layered architectural pattern and as it can be seen below in the logical architecture chapter, the system is divided in several layers where each layer communicates with the layers below and above it.

First, we present the general overview and the logical architecture and finally, the physical architecture.

3.4.1 General Overview

This section pretends to describe the general overview of the architectural model in terms of packages and layers. The next figure shows the packages that make the logical view of the system's architecture.

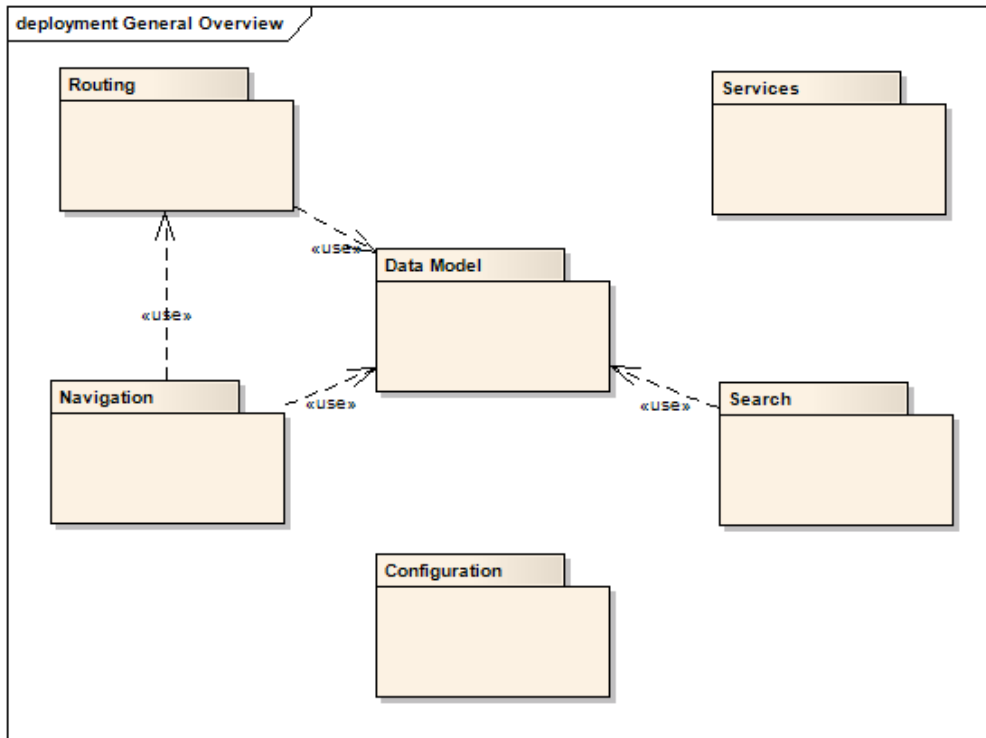


Figure 3.1: General Overview of the Architecture

Description of the packages in the figure:

Data Model: Package which contains all the information regarding the map and different networks as well as all the information regarding transports and pedestrian enhanced navigation. This information has its origin in the database.

Navigation: Package with the classes that allow the user to visualize the information contained in the data model and navigate through the map.

Routing: Package with the classes responsible for planning, managing and monitoring the calculated routes.

Search: Package with the classes responsible for searching the map for some particular information the user wants to find, like addresses, POIs, schedules for a particular transport at a particular stop, among others by querying the data model and other databases.

Configuration: Package with classes responsible for saving and editing the settings of the application as well as the managing the profile of the user.

Services: Package with classes responsible for managing and retrieving information from services provided by external entities such as transport real-time delay or sharing favourite routes and POIs in a community.

3.4.2 Logical Architecture

In this section we present the logical architecture of the system. The logical architecture has the objective of depicting the division by layers/implementation concepts.

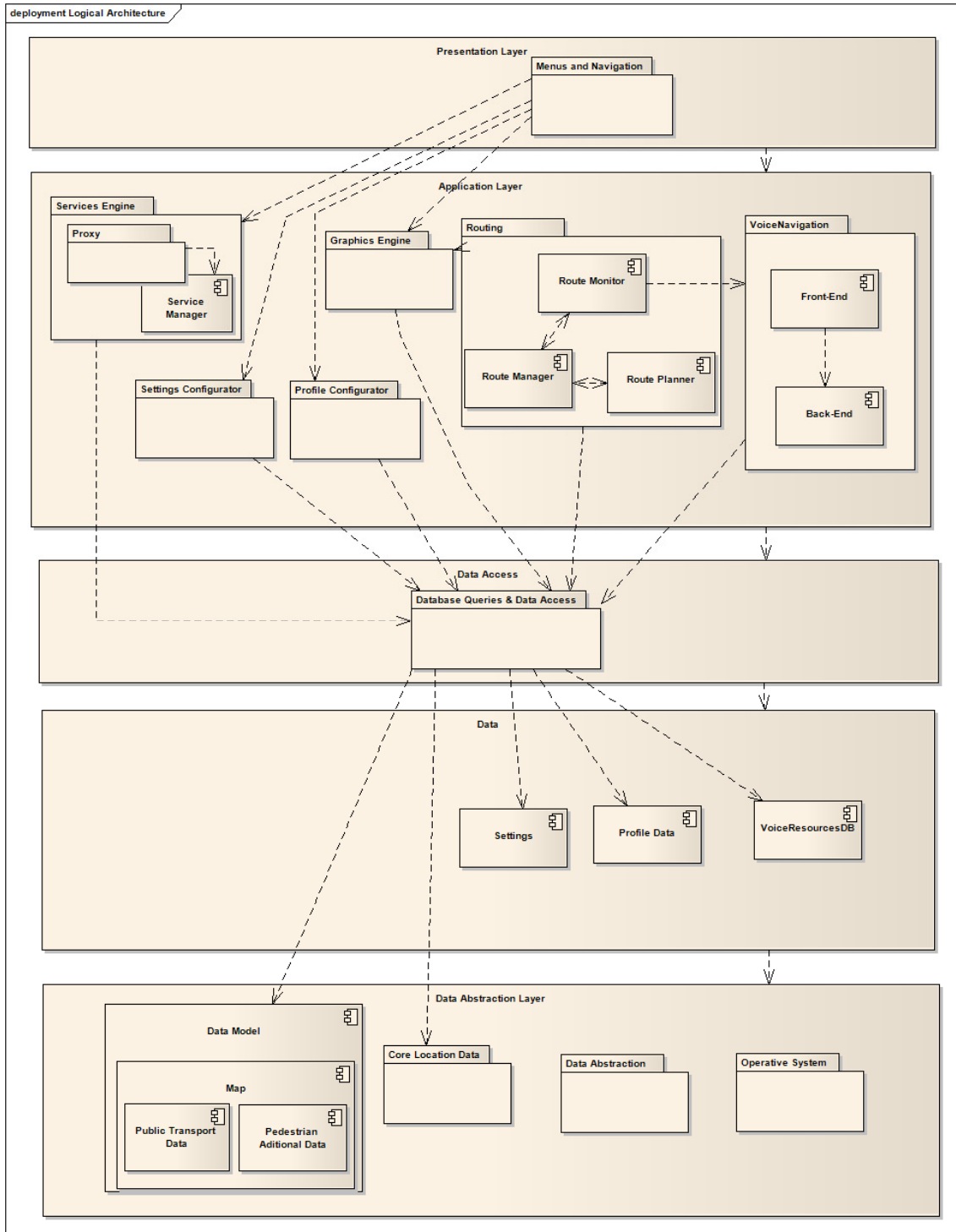


Figure 3.2: Logical Architecture

As it can be seen the system is divided into 5 layers:

- Presentation Layer
- Application Layer
- Data Access Layer
- Data Layer
- Data Abstraction Layer

The **Presentation Layer** is the layer that performs the connection with the user. It is responsible for presenting the various interfaces and menus of the system as well as displaying the map for navigation, displayed above as the **Menus and Navigation** package.

The **Application Layer** is where the services and functionalities provided by the system are processed. In this layer we can find the **Graphics Engine** package, the **Routing, VoiceNavigation, Services Engine, Profile Configurator** and **Settings Configurator** packages. The Routing package is where the components responsible for planning, managing and monitoring routes are located. These components communicate with the graphics engine for visual representation of the data representing the route(s); with the VoiceNavigation package where the audio navigation aids are processed to support the user along his route; and with the Data Access Layer's Database Queries & Data Access package, whose function is explained in the Data Access Layer. The Settings Configurator package is where the classes responsible for managing the options provided in the settings menu are contained. Among the settings configured by this package are routing options such as the routing mode, services options and display options. In the Profile Configurator package we have the profile settings manager as well as the profiler component which studies the history of the user's choices in routes and routing options and provides the user with the option to change his profile to one which the profiler determines more adequate. This is done via a rule-based engine. When changing to another profile, the system changes the certain settings to the profile type selected. For example, if the user is in a different city from where his home is defined and he is choosing routes with touristic POIs as destination, as well as using public transport routes as his means of travelling the city, then the user may be a tourist and so the system may suggest switching to that profile and with this change alter some definitions to better suit this profile.

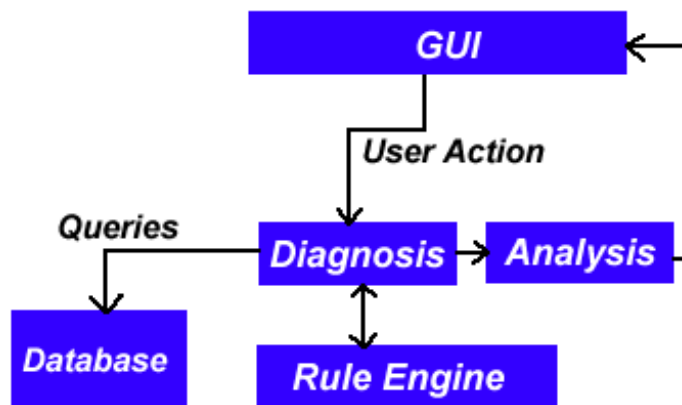


Figure 3.3: Profile Configurator

The Profile Configurator can be implemented in Rule Markup Language (RuleML) [RUL10] and since this is markup language based in XML it can be updated and maintained without much trouble through Web Services via the Services Engine [HEC03], [GOI07].

Finally, the Services Engine is the package responsible for managing the services available, the connections to these services as well as parsing the information retrieved. Examples of such services are for instance the real-time delay of some public transport on a particular run or the update of the timetables for a certain line of a public transport type. In order to retrieve information from a particular service, the system needs to have an Internet connection either by Wi-Fi or over the carrier network. The Services Engine is composed by the Service Manager component and the Proxy package which contains the proxies to the services installed. The Service Manager component, like its name suggests, is responsible for managing the services added to the system.

The **Data Access Layer** is responsible for performing queries to the various databases, such as the VoiceResourcesDB, as well as performing CRUD (Create, Read, Update, Delete) operations on the databases. These classes are represented as the **Database Queries & Data Access** package.

The **Data Layer** is where all the resources that are platform dependent are kept. This includes databases that are particular to a device, such as the VoiceResourcesDB component where the phonetic transcriptions for voice synthesis are saved, as well as configuration and profile settings, Settings and Profile Data components respectively.

At the bottom of the logical architecture is the **Data Abstraction Layer** which is where all the data that is platform independent is kept. This layer is a very important one if we are to have a system which is portable and with its core being platform independent. Examples of such abstract data are the abstract data types and structures represented in the **Data Abstraction** package; the **Core Location Data** package where all the information regarding location, such as latitude, longitude, speed, altitude, if the GPS has a fix or not and others, is saved. Also, it is in this layer that the map database and data model information are kept as it can be seen in the figure above represented by **Data Model**, **Map**, **Public Transport Data** and **Pedestrian Additional Data** components.

3.4.3 Physical Architecture

In this section we describe how the physical network of the system is configured, as well as how the different modules connect to each other.

Since the navigation application is pretended to run in a mobile device, most of its functionalities are performed locally. So, only the services that are externally provided need an Internet connection via Wi-Fi [BAR06] or through the carrier network in order to be able to establish a connection to the server providing that particular service and retrieve the necessary information.

A Location-Based Solution

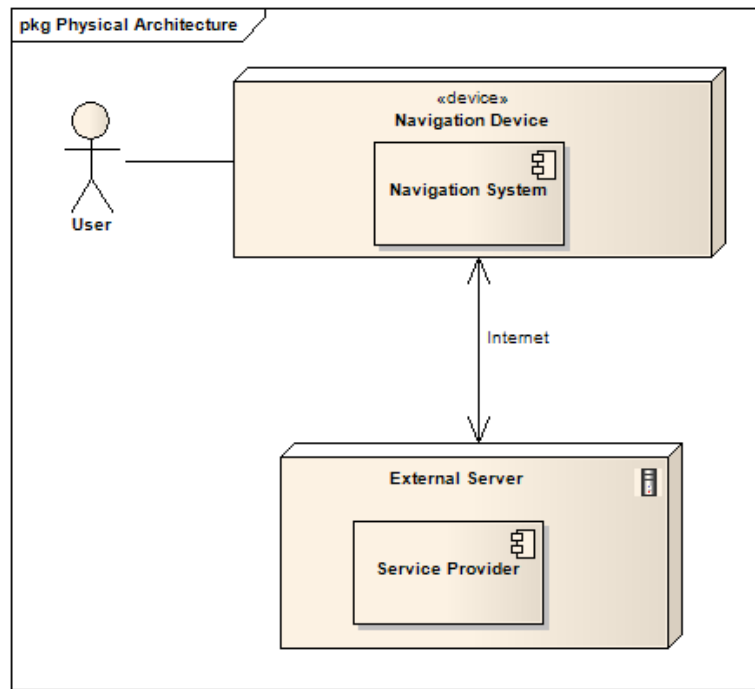


Figure 3.4: Physical Architecture

So we have the following components on the physical architecture:

- **Navigation Device:** where the navigation application is running;
- **External Server:** the server where the information for the particular service is provided from;

In the navigation application we have the Services Engine which connects to a service through the proxy (meaning the service client proxy needs to be added to the application in order to use the service and parse the information retrieved).

3.5 Chapter Summary

In order to have a starting ground for the specification of a solution for public transport users of pedestrian navigation systems, the specification of the requirements for these information systems was needed. So, through user stories, in this chapter it was presented the various requirements for the users of these systems when regarding functionalities that the system should have.

Nowadays, one of the biggest limitations for pedestrian navigation systems is the limited capacities of the mobile devices and with their evolution, new opportunities for these systems to evolve appear as well. In this chapter it was proposed an architecture for such a system that complies with the pedestrian user's requirements and keeps a level of abstraction in order to be as portable as possible between different mobile devices. The proposed architecture is a typical layered architecture that contains the following five layers: Presentation Layer, Application Layer, Data Access Layer, Data Layer and Data Abstraction Layer. The Data Abstraction Layer is the key to provide the previously mentioned portability.

The Presentation Layer is responsible for the presentation of the menus and navigation interfaces for the user. The Application Layer is where the main functionalities of the system are defined such as the Services Engine responsible for the management and connection to the various services; the Profile Configurator, responsible for configuring and updating the user's profile; the route Planner, Monitor and Manager responsible for planning, monitoring and managing the routes calculated by the system; and the Voice synthesizer's front and back end for assisting navigation with audio instructions. The Data Access Layer is the layer responsible for performing queries on the various databases. The Data Layer contains the information that is device dependent and the Data Abstraction Layer contains information that is device independent.

Chapter 4

Data Model

4.1 Introduction

The data model is one of the core components of a navigation system. The data model contains all the information required to create the map and its several layers which represent the hierarchically defined levels of detail of the navigational network graph [ARC10]. Also, all the information that will represent the buildings, areas, POIs and all information detailing these points is contained inside this part of the system. Therefore, its importance is very clear when defining a solution for an information system for public transport users. Since one of the objectives of this dissertation is the definition of a spatiotemporal data model, one such model will be presented in this chapter.

As mentioned before, by having a higher degree of freedom, pedestrians can move through almost everywhere on the real world. Such level of detail is still impossible in the current existing models but such a high degree of freedom is not considered necessary. Nonetheless, it is important for the pedestrian to be able to have represented on the map of a navigation system some components that are still missing in today's models. Information about crosswalks, bridges over streets or tunnels below them, crossable areas by pedestrians only like parks, squares or parking lots, as well as stairs, ramps, escalators and elevators and sidewalks are examples of important information for the pedestrian. Aside from the pedestrian navigation information there also the need to have information about public transport present. Information about the public transport companies, their several systems, lines, stops and station locations, each and every run performed through those stops of a particular line and the schedules for those runs are also examples of information that needs to be considered by the data model. Linked to the public transport there are zones which need to be defined, since the definition of most of the public transport system's tariffs depend on zones. So, with the inclusion of zones in the data model, it is possible to include detailed information about the tariffs and cost of the public transport themselves.

Another possible type of information to be added to the data model is information about buildings and their interior constitution. Definition of floors, areas, items and access points make interior modeling of buildings for indoor navigation possible to be defined.

Typically and as mentioned before, the several parts of information that constitute the data model can be arranged in packages of information. Like many examples of data models available that can be seen on several projects on the internet. The following data model is

divided into **PedestrianInformation** package, **PublicTransport** package, **MapInformation package**, **Costs** package, **Zones** package and **IndoorNavigation** package. In the PedestrianInformation package is contained all the information regarding pedestrian-only navigable areas such as the crosswalks and sidewalks. The PublicTransport package contains the information regarding the composition of public transport systems and schedules. The MapInformation package is where the core information about the map such as nodes and edges is contained. The Costs package contains information about the tariffs of the public transport and the Zones package information about public transport's zones. Finally, the IndoorNavigation package contains the information about the components of buildings for indoor navigation purposes. Other packages can be found such as the **ProfileSettings** package where the information about the definitions of profiles is kept and the **History** package where the history about the users' choices in routing and navigation is kept.

In order to achieve a model with a good quality there are a few rules that it has to follow. According to West and Fowler [WES99] in order to develop a high quality data model, the data model must:

- meet the data requirement;
- be clear and unambiguous to all (not just the authors);
- be stable in the face of changing data requirements;
- be flexible in the face of changing business practices;
- be reusable by others;
- be consistent with other models covering the same scope;
- be able to reconcile conflicting data models.

Also, they must obey a series of principles:

1. Candidate attributes should be treated as representing relationships to other entity types.
2. Entities and records should have an internal identifier within a database or exchange file. These should be artificial and managed to be unique. Relationships should not be used as part of the internal identifier.
3. Activities and associations should be represented by entity types (not relationships or attributes).
4. Relationships (in the entity/relationship sense) should only be used to express the involvement of entity types with activities or associations.
5. Entity types should represent, and be named after, the underlying nature of an object, not the role it plays in a particular context.
6. Entity types should be part of a sub-type/super-type hierarchy of generic entity types in order to define a universal context for the model, and to avoid duplication of concepts and data.

Data Model

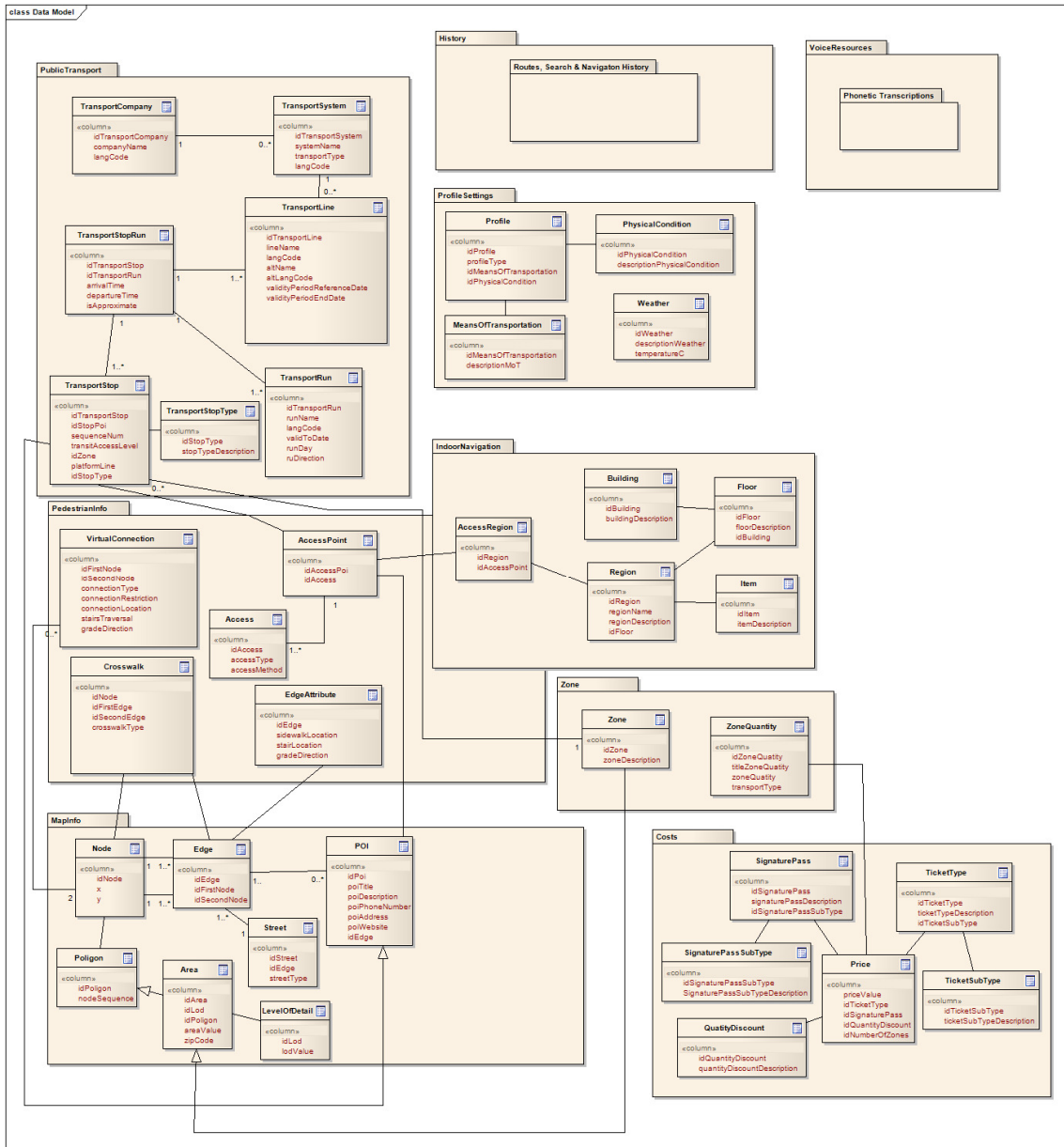


Figure 4.1: Data Model

4.2 Pedestrian Information

It has been said before that there is more information for the pedestrian navigation purposes that needs to be added to the data model in order to create a more suitable and richer navigation experience for the pedestrian user. There is a need to introduce crossable areas for pedestrians only like parks and squares, crosswalks, bridges and tunnels for pedestrians to cross the streets and highways safely and sidewalks for pedestrians to circulate safely through the streets. So, in order to add this information to the data model we have to divide it into different parts that represent different types of information. For instance, we have the crossable areas such as parks and squares, VirtualConnection; we have crosswalks such as your ordinary parallel or zebra crosswalk to cross the street or a bridge to cross over a highway, Crosswalk; and then you have specific information regarding access to public transit POIs, AccessPoint, which are the stations platform's or bus stops for example; and finally, there is additional information comprised in a set of link attributes which concern sidewalks and stairs on the sidewalks, EdgeAttribute, and can be used for map display and for pedestrian routing.

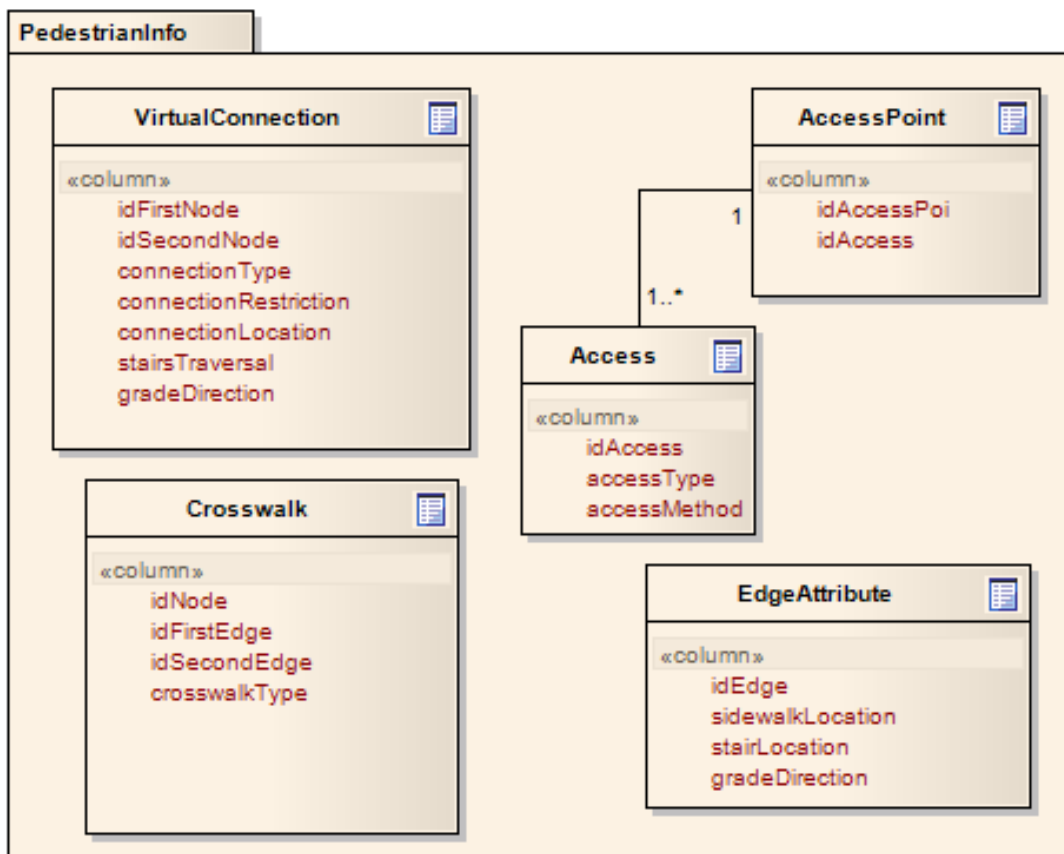


Figure 4.2: Additional Information for Pedestrians

EdgeAttribute

We will start by describing in detail the EdgeAttribute table. This table adds to the data model new information to enrich the pedestrian experience, either visually at the map display level or both at the map display level and for pedestrian routing purposes as well. We can identify the edge where there is new information to add in the idEdge attribute which is a both foreign and primary key to this table. By inserting an existing edge ID we can associate it with sidewalkLocation, stairLocation and gradeDirection. The sidewalkLocation attribute identifies if a sidewalk is present along that particular edge and at which side or sides it exists by taking in one of the following values: LEFT, MIDDLE, RIGHT, LEFT_MIDDLE, MIDDLE_RIGHT, LEFT_RIGHT, LEFT_MIDDLE_RIGHT or NONE. When there are no sidewalks present along the edge the value NONE is coded. The stairLocation attribute is used to indicate if there are stairs present in reality along that particular edge and can be used as well for map display and/or pedestrian routing. The valid values for this attribute are the same as those for the sidewalkLocation: LEFT, MIDDLE, RIGHT, LEFT_MIDDLE, MIDDLE_RIGHT, LEFT_RIGHT, LEFT_MIDDLE_RIGHT or NONE. When there are no stairs present at the edge the value NONE is coded. When the values for both the sidewalkLocation and stairLocation attributes are equal, for example LEFT and LEFT, this means that there are stairs in that particular sidewalk. If we are in a pedestrian only edge where there are stairs, then we code the stairLocation with the value MIDDLE. This information can be used to avoid routes with too many stairs and therefore avoiding the creation of problems for handicapped people for example. The gradeDirection attribute is only applicable to stairs and it indicates the direction in which the grade is changing, i.e., UPWARDS when it is rising towards the reference node or DOWNWARDS when it is decreasing towards the reference node. Once again, this attribute can be used for map display and for pedestrian routing as well.

VirtualConnection

Next, we will describe the VirtualConnection table. This table has the information regarding connections between nodes which are crossable only by pedestrians, like for example a park. There are no edges that represent paths for the pedestrians to walk on but nonetheless the path physically exists because it is crossable by the pedestrian. Therefore we create virtual edges or virtual connections which represent these paths. For example, in park with 2500m² there are paths that cross the park that pedestrians take. So, we can create a path between the nodes which are in the diagonals of the park area and therefore make a connection that can be used not only for map display but for routing purposes as well by considering the path which crosses the park and making it a more viable and realistic solution than telling the pedestrian to cross the park by walking around it instead of through it.

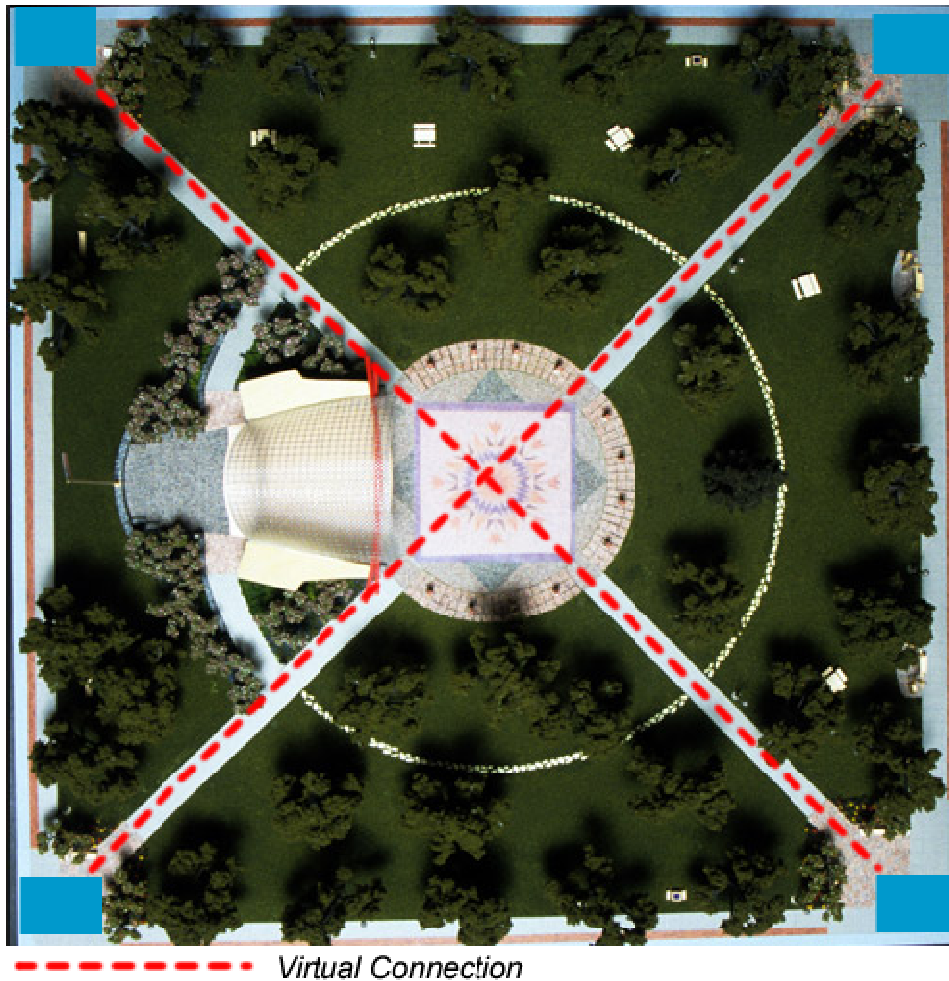


Figure 4.3: Virtual Connection through a park

In this table we then have the following attributes: `idFirstNode`, `idSecondNode`, `connectionType`, `connectionRestriction`, `connectionLocation`, `stairsTraversal` and `gradeDirection`. The first two attributes, `idFirstNode` and `idSecondNode`, are references to the nodes at which the virtual connection is going to be created; the `connectionType` attribute indicates the type of the virtual connection that is being created between the two nodes and can take one of the following values: `ESCALATOR`, `ELEVATOR`, `GROUNDLEVEL`, `PEDESTRIAN_RAMP` or `STAIRS`; the `connectionRestriction` attribute indicates if there are any restrictions to the access on the virtual connection being created. Seeing that this virtual connection can be a connection that unites two nodes at the edges of a park, for example, this park can be closed at night and therefore meaning that the pedestrian cannot cross it during the night and only during the day. This attribute can take one of the following values: `DAY_RESTRICTION`, `NIGHT_RESTRICTION`, `NONE`. The next attribute on the table, `connectionLocation`, is related to the `connectionRestriction` attribute and indicates if the location of the virtual connection created is through a building, a free park or one where there is a fee, a plaza or a street and therefore it can take one of the following values: `BUILDING`, `PARK_FREE`, `PARK_FEE`, `PLAZA` and `STREET`. The `stairsTraversal` attribute is a boolean attribute and indicates whether there are stairs, `TRUE`, or not, `FALSE`, in the virtual connection. We can use this information when calculating a route for the pedestrian making it so that routes

with stairs are avoided, for example. The final attribute in this table, grade direction is related to the connection types ESCALATOR, PEDESTRIAN_RAMP and STAIRS and as referred before in the EdgeAttribute table, indicates the direction in which the grade is changing, i.e., UPWARDS when it is rising towards the reference node or DOWNWARDS when it is decreasing towards the reference node and this information can be used for pedestrian routing and for map display.

Crosswalk

Crosswalks are important in the data model because besides assuring a better representation of the pedestrian's reality and therefore a more realistic experience they also contribute for a more secure navigation for the pedestrian by giving him directions for safe crossing of streets, highways and such. In order to add crosswalk information to the data model we need to indicate three things which locate the crosswalk in the map and the rest of the attributes describe the crosswalks type and other additional information. The first three attributes mentioned before are the node around which the crosswalk will be represented, idNode, and the two edges that are connected to the node, idFirstEdge and idSecondEdge and which when derived indicate the "corners" from which the crosswalk starts and ends. The order for these two edges is indifferent and works equally if you specify idFirstEdge and idSecondEdge or idSecondEdge and idFirstEdge. Two edges and a node are needed to code a crosswalk because there is a need to have two points around a node from which to connect the crosswalk, i.e., a starting and ending point for the crosswalk around a particular node. To achieve these points, instead of adding two additional nodes to connect the crosswalk and therefore increasing much significantly the amount of information in the map database, we can use the information that is already there and derive these points from this information. The information from which this derivation is done is the previously referred node and its two connecting edges. So, using the node and one of the edges, if we rotate the edge identifier clockwise we get one of the "corners" around the node where the crosswalk physically exists. We do this for each edge that connects to the node and get the two points to connect the crosswalk to, the starting and ending point of it. These "corners" are not attributes of the data model but are derived from the node and edge's specification.

Finally, there is the crosswalk type attribute, crosswalkType, which describes the type of crosswalk that is being coded and its possibilities are: SOLID, PARALLEL, CONTINENTAL, DASHED, ZEBRA, LADDER, BRIDGE and TUNNEL. All except BRIDGE and TUNNEL refer to normal crosswalks types. BRIDGE and TUNNEL are also considered crosswalks because they are for pedestrian use and serve the purpose of crossing a highway for example, either from above or from below, respectively.

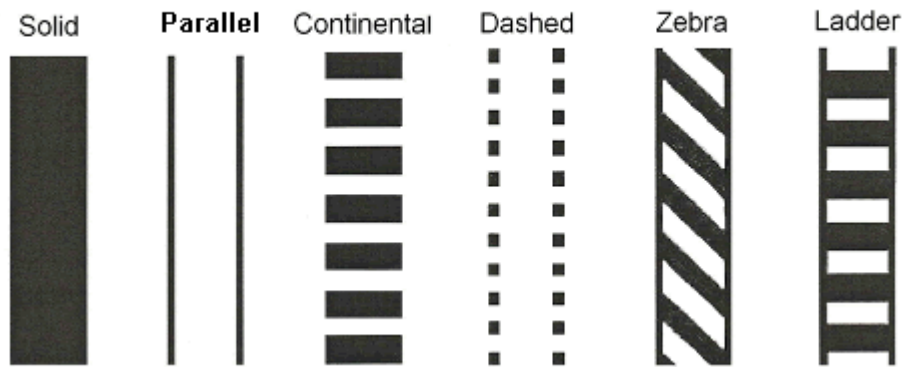


Figure 4.4: Crosswalk Types

Examples:

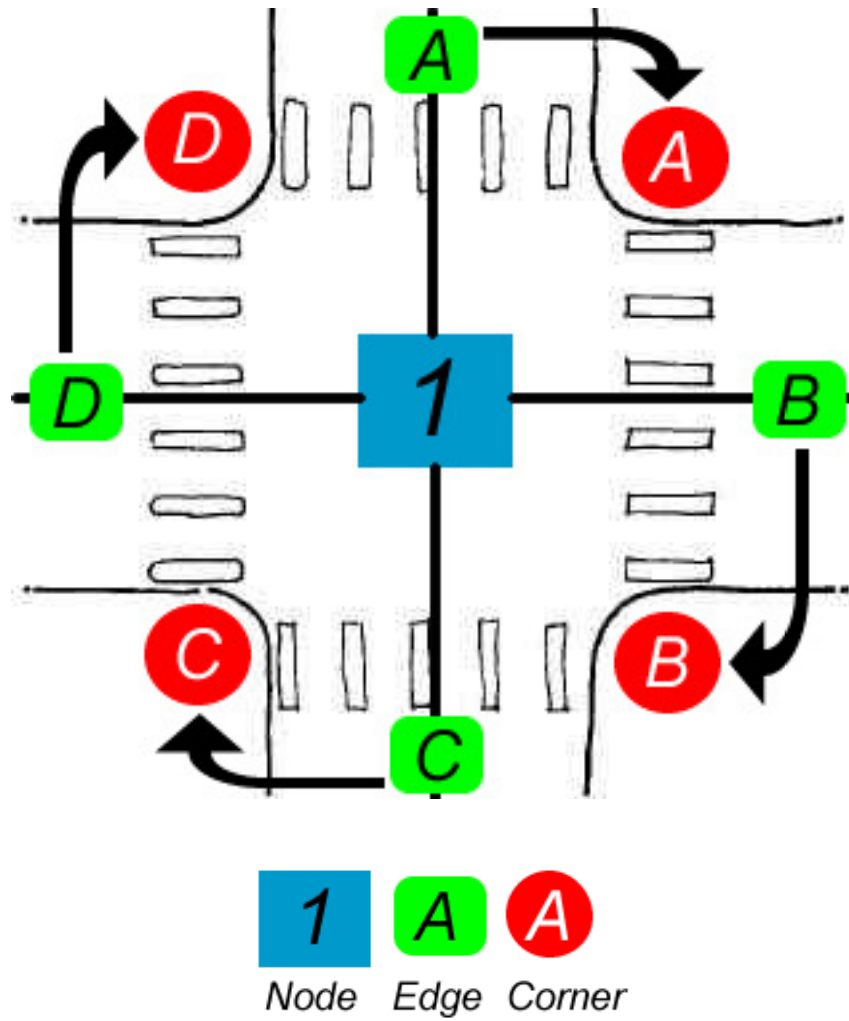


Figure 4.5: Corner Points Derivation Mechanism for Crosswalk Definition

For a better understanding of the how the crosswalks are coded we will demonstrate the procedure with the help of the figure above. The figure above depicts an intersection at Node 1 and connected to Node 1 are Edges A, B, C and D. As it can be seen above, the crosswalks start and end at the corners of the node. So, if we rotate the edge identifiers clockwise we get the corners A, B, C and D which are represented as the red circles. For this node we would have the following values for the crosswalks:

Table 4.1: Crosswalk Table Example

idNode	idFirstEdge	idSecondEdge	crosswalkType
1	A	B	CONTINENTAL
1	B	C	CONTINENTAL
1	C	D	CONTINENTAL
1	D	A	CONTINENTAL

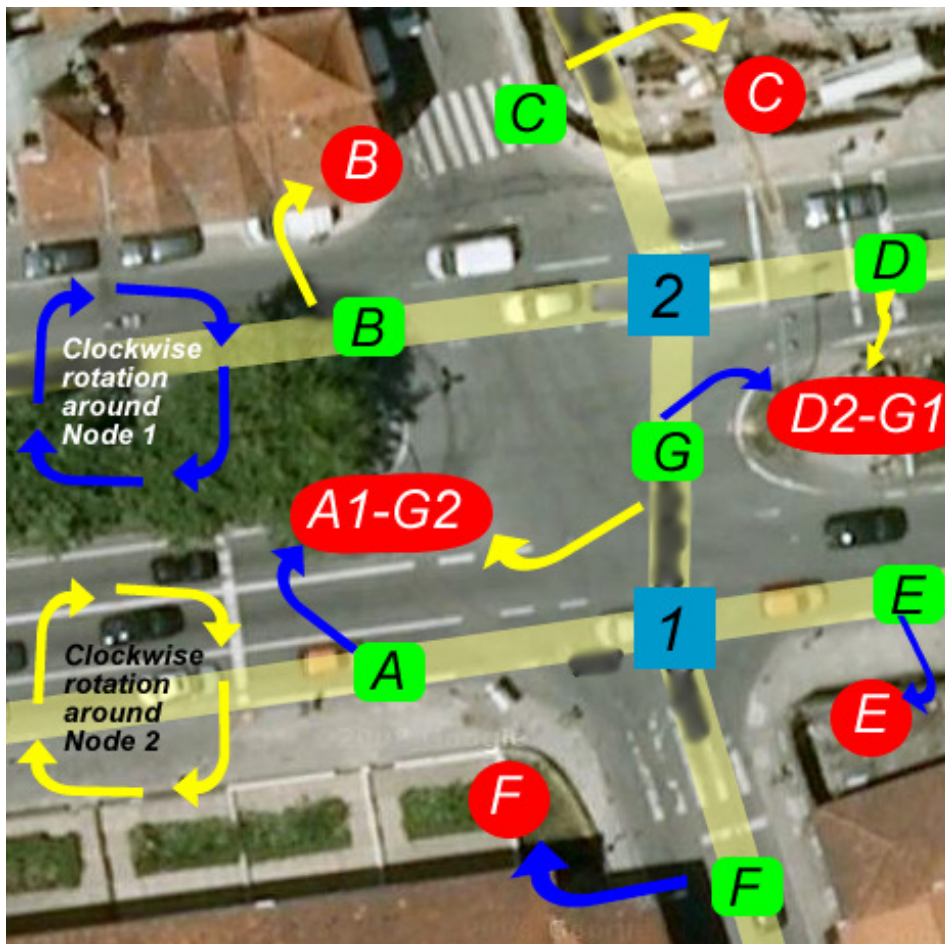


Figure 4.6: Crosswalk Complex Example

This example is a little more complex than the previous one mainly because it deals with two nodes which share one equal edge, edge G. The mechanism is still the same, to derive the corner points of the node from which the crosswalk starts and ends. We rotate clockwise around each node, for each edge and get the corners A1, G1, E and F for node 1 and corners B, C, D2 and G2 for node 2. Corners which are derived from a shared edge between two nodes and represent a point along the crosswalk take the number from the node which they are derived from, i.e., G1 is derived from edge G and node 1 and G2 is derived from node 2 and edge G. When coding the values for the edges on the table the number is omitted and as such the result from a situation like the one represented above is the following:

Table 4.2: Crosswalk Definition Table for Complex Example

idNode	idFirstEdge	idSecondEdge	crosswalkType
2	B	C	CONTINENTAL
2	C	D	PARALLEL
1	G	E	PARALLEL
1	E	F	CONTINENTAL

In the cases where the virtual connections are of the TUNNEL or BRIDGE type we get a line in the Virtual Connection table like the following one:

Table 4.3: Crosswalk Definition Table for the BRIDGE type

idNode	idFirstEdge	idSecondEdge	crosswalkType
1	A	B	BRIDGE

Table 4.4: Crosswalk Definition Table for the TUNNEL type

idNode	idFirstEdge	idSecondEdge	crosswalkType
1	A	B	TUNNEL

There is a need to place a node in that position so that a crossing can be coded in the same position.

AccessPoint and Access

The AccessPoint and Access tables comprise a set of attributes that refer to the access from the public transport related POI to that particular public transport vehicle. We are considering and representing the stops along a line of a public transport as POIs and so, the AccessPoint table indicates which POI the access is being indicated, idAccessPoi, and links to the Access table through the idAccess column of the table. The Access table has the ID value for the Access, idAccess, the access type, accessType, and access method, accessMethod, for a particular POI. The accessType attribute indicates whether the access is entrance only, exit only or both entrance and exit and so it is coded with one of the following values: ENTRANCE, EXIT, ENTRANCE_AND_EXIT. The accessMethod attribute indicates the method of access that is present at the POI, for example an escalator. It uses one of the following values: ESCALATOR, STAIRS, ELEVATOR, PEDESTRIAN_RAMP, LEVEL or NONE. The LEVEL value indicates that the access point is at the same level as the public transport. These two tables exist because there is the possibility of existing more than one method of access to the public transport present and so, more than one access type and method is needed for each access point. By separating them into two tables we prevent redundant information.

4.3 Public Transport Information

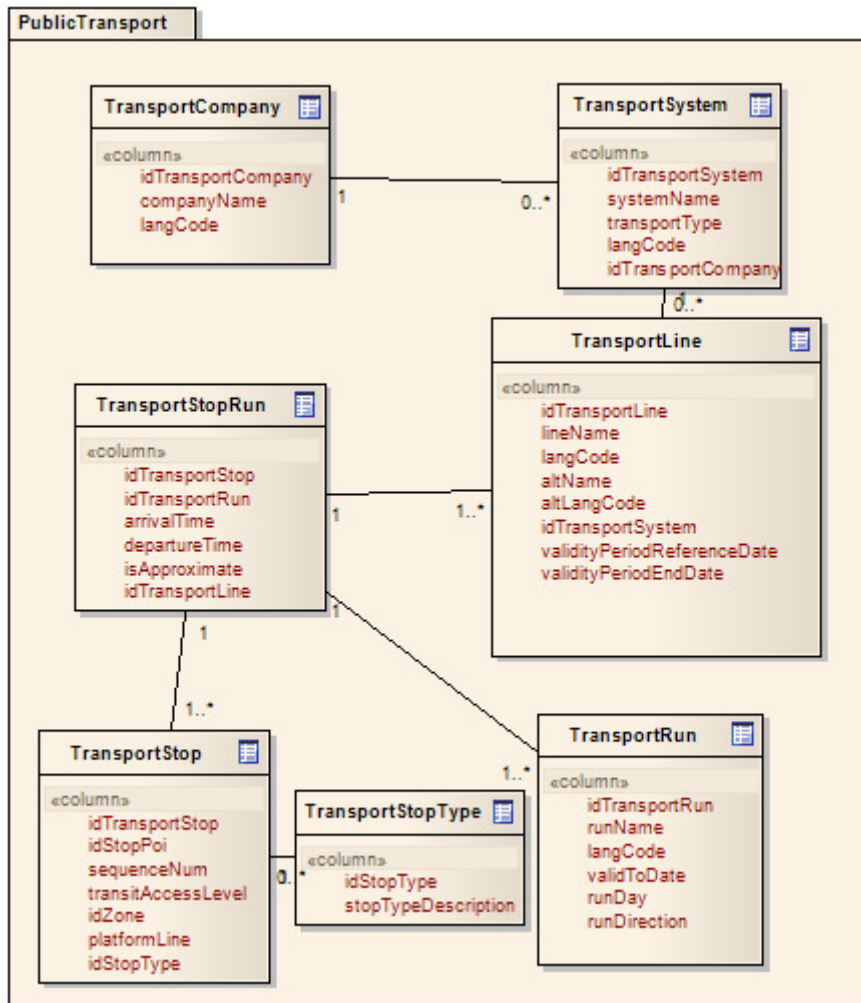


Figure 4.7: Public Transport Information

As public transport information that is regarded in the data model, we can divide it into 7 tables. TransportCompany, TransportSystem, TransportLine, TransportStopRun, TransportStop, TransportRun and TransportStopType. TransportCompany is where information regarding the public transport company is saved such as the company name, telephone number, the head office's address and other information regarding the company. TransportSystem is the table regarding the various transit systems that are part of the company, for example, bus systems, tourism bus systems, subway systems and train systems. A company can have more one or more transport systems so, besides the name of the system, the transportType is required to identify the transport type. TransportLine is the table that concerns the general information about each of the several lines of one system from the company. Each transport system can have one or more lines. The information about the transport line contains the line name which can be for example a code number like line 600, an alternative name to the line, altName and the validity start and end periods for the line, validityPeriodReferenceDate and validityPeriodEndDate. Aside from the general information about the line, each line has two other types of information, TransportStop, which concerns all information about the set of stops

Data Model

that make a transport line and TransportRun, which contains the information about each run that the transport makes throughout the all the stops that compose the line. In the information regarding each run there is the identifying name for the run, runName, the validity period for the run, ValidToDate, the day which the run is on, runDay and the direction of the run, runDirection, to determine if its going to its destination or coming to its origin. The information in the TransportStop table contains a reference id for the stop, idTransportStop, the node to which it is associated, idStopPoi, the number of the stop in the sequence of stops for the run, sequenceNum, and the transitAccessLevel which represents the information about the location of the platform or street where the stop is located and contains three possible values, ABOVE_STREET_LEVEL, ON_STREET_LEVEL or BELOW_STREET_LEVEL. Another value that this table contains is the zone identifier, idZone which is used for verifying the costs of a travel. It may contain as well as an optional value the platformLine number in case the stop is a station with more than one platform. There is also the stopType to identify if the stop is of station, normal, platform or interchange type, defined in the TransportStopType table. TransportStopRun is the table that links these two sets of information together, the stops and the runs, with the transport line that they belong to. Also, it contains the information about the times for each run the public transport makes on that line and among this information there is the frequency of the public transport on that line in each stop, and the arrival and departure time for at each stop and at each run. The langCode and altLangCode columns of the tables represent the language code in which the information is detailed, for example, EUA or POR.

For each transport system, lines are defined by a set of stops and each stop is represented as a subtype of a POI. In order to connect these special POIs, a subtype of edge is used which has the type of the transport, for example, rail or street. Using these subtypes, a network is constructed and each network will have a different value for the hierarchy in the graph that represents the whole network. The several networks are then connected by the stop POIs on the interchange stops.

4.4 Additional Information

Costs

There is more information that can be added to the data model which concerns both public transport and pedestrian users, information about the travel costs for the pedestrian for a particular route. Adding this information can give the users some useful insight into how much they will expend with travel expenses making it primarily destined and most useful to tourists. Despite the fact that this information is useful, in the end it is not crucial information for the pedestrian users of a multimodal navigation system as appointed by Grotenhuis et al. [GRO07] in their study. Although not being crucial, if present it is more information that the users have about the transport systems and they can consult it to see if a particular signature pass would be more adequate for normal and regular route, allowing them to save some money for example.

The cost information is divided mainly into two groups: Zone Information and Travel Costs Information. Zone information concerns mainly information about travel zones and it is not only used for costs. The Zone table identifies the public transport zone that the stop is part of. The ZoneQuantity table indicates the zone quantity or number of zones which one ticket type will be valid to cross through and the transport type associated. As for the information on the costs package, there are 6 main tables. The SignaturePass table contains the definition for signature passes such as the name for the signature pass which acts as the main type of signature pass, signaturePassDescription and the sub type of signature pass contained in the SignaturePassSubType table. The same thing happens for regular tickets on the TicketType and TicketSubType. The quantity discount table defines special values for promotions on tickets bought in a larger quantity. Finally, the Price table, based on the ticket type or signature pass (because one of them is required), along with the quantity discount (which is optional) and the number of zones traversed, defined in the ZoneQuantity table (required) defines a price value for the ticket.

Indoor Navigation Information

With the increase of interests in interior cartography rising, one of the logical next future steps for navigation systems is indoor navigation. In order to be prepared for indoor navigation it is needed to add to the data model some additional information regarding buildings and their schematics [REH05]. In order to provide a possible navigation on buildings' interiors it was defined in the data model buildings, floors, regions, items and access points to regions. The Building table contains the general information about the building such its description and type. The floor table contains information about each of the floors that make a building. A floor is composed by regions. These regions can rooms or entire hallways and they are a subtype of area as well. A region contains items, which can be obstacles or signals. Finally, to make the connection between regions there is the AccessRegion table which links together a region and an AccessPoint. In order to prevent redundancy in the information, the access point tables from the pedestrian information package can be used. So, the types for access points can be stairs, escalators or elevators.

Just like in the outdoor navigation, an indoor topology formed by nodes and edges still needs to exist in order to define the walkable paths for the user. Indoor maps would have to be defined as the lowest level in the level of detail of map in order to be in the lowest level of the hierarchy of the graph. In order to connect the interior topology to the outside, it could be defined a node at the entrance of the building with an entry on the AccessPoint table marking an access to a building. Another option for when the building has two or more entry points is to defined a node at each of the entrances and define a virtual connection with the connectionLocation coded with BUILDING and just like before define each node as an access point.

Profile Information

The profile information which is kept in the ProfileSettings package of the data model contains the data regarding profiles and profile types. This information is the one that would be used along with the information contained in the History package to determine through the Profile Configurator component mentioned before, the appropriate profile for the user. With this change in the profile, the system would change the settings of the application as well so as to present the user with a more personalized experience. All these options can be manually defined by the user in a profile settings menu.

Among the information contained in this package there is the Profile table which contains the main definitions of the profile such as the type and description of the profile. There is the PhysicalCondition table which contains data about the physical condition of the user. Among the options in this table are handicapped, fitness and aged. Also, there is the MeansOfTransportation table which defines the means of transportation for the user, namely feet or wheelchair for example. There is also the weather table which can be used along with a web service to retrieve weather information for the user's location on that particular day and saved in this table in order to be taken into account when calculating routes. With this information the route can be calculated so as to avoid big walking distances as much as possible or streets with pedestrian ramps and stairs for example, in order not to tire the user as much as possible, especially if he is an older or handicapped person.

4.5 Chapter Summary

In this chapter it was proposed a data model that can extend a previous existing data model by adding information considering pedestrian navigation and public transport information. In order to achieve a data model that can support and comply with all the requirements from the pedestrian users of the information systems, the data model needs to contain a large amount of information. Among this information is not only the previously mentioned pedestrian and public transport information but other information that can be useful for pedestrian navigation and to make the user's experience a better and more personalized one.

Information about costs and zones are an example of such information that, despite not being the most important information for the pedestrian user can facilitate his travel in order to know how much he is going to need to expend and on a next level, provide a way to find and provide a route with the lowest expense for the user. Previewing a future trend of indoor navigation for pedestrian navigation systems, it is provided a way to add information about the interior of buildings in order to use it for indoor navigation as well as a possible way to link them to the rest of the map, the outside map. Finally, in order to make the user's experience a more personalized one, the information regarding the creation of profiles to identify the main preferences of the users and the different types of users was added to the data model.

Chapter 5

Sensor Enhanced Navigation (Magnetometer and Accelerometer)

5.1 Introduction

Many of the most recent mobile devices have integrated sensors like accelerometers and magnetometers. The inclusion of these sensors in the mobile devices opens doors to new possibilities and ways to fix some existing problems in pedestrian navigation systems and/or improve the experience of the user of these systems [RET04]. First, since the pedestrian moves slower than any motorized vehicle, he has a different perspective and better awareness of his surrounding environment. Therefore, his spatial resolution is different because he can observe the environment in a more detailed way and that causes the necessity to have different navigation information than that for vehicles. Secondly, since pedestrians can guide themselves better using landmark information instead of left or right information and sometimes assess distances badly, instructions like “turn left at 300 meters” are not the best way to present the navigation instructions to the pedestrian user [MAY03], [DEN01], [ELI04].

Using the magnetometer sensor it is possible to capture information about the heading of the mobile device much like a digital compass. Using this captured heading information and adding it to the Core Location Module of the navigation system, one can include the heading information in several other modules, like the graphical interface, and represent the heading at which the user is currently oriented to. This way the user knows which direction he is facing at the current moment and the change in heading is immediately visible in the interface even if there is not a GPS position update. The experience of the user is improved since he is now able to determine which direction he is facing at any moment, much like what happens with the Google Maps application for the iPhone. In the figure below we can see the current orientation through the direction of the cone.

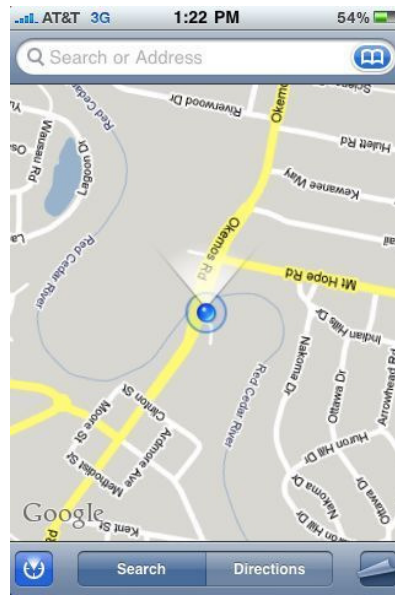


Figure 5.1: Compass view for the iPhone Google Maps

With the inclusion of data relative to the current heading from the magnetometer sensor the user has the possibility to know which direction he is facing at any given moment. Nonetheless, the low speed at which the pedestrian moves is still a problem. Here is where the accelerometer sensor data comes in. Adding the information relative to the acceleration of the mobile device gives the system the possibility to know when the device is moving. However, there are some problems associated with the accelerometers that are normally present in mobile devices. First, they are sensitive to the slightest variation in acceleration, wherever it comes from moving the phone in a determined direction or from just tilting it. Second, there is always a major force at work, the force of gravity. Gravity is the acceleration caused by the gravitational force on any object with mass. In order to solve these problems it is advisable to implement a linear filter. The most relevant linear filter for a navigation system is the high pass filter which passes high frequencies well but attenuates the frequencies lower to the filter's cutoff frequency. This cutoff frequency is a design parameter of the filter, meaning that the amount of attenuation for each frequency is customly defined. So, it is defined a value that attenuates the values correspondent to the gravity acceleration in order to use only input from acceleration occurring when moving the device. More details about the implementation will be presented in the next chapter.

5.2 Implementation

The implementation and integration of the data from the two sensors, the magnetometer and the accelerometer, to the Core Location Module of the navigation system for use in the pedestrian navigation mode was performed at the company NDrive. This integration was performed using an iPhone 3GS and the NDrive navigation application for the iPhone. The implementation at the Core Location Module was coded in C++ and at the functions for activation and gathering of the inputs from the sensors was coded in Objective-C.

As mentioned before we gathered input from two sensors. The integration with the magnetometer was performed first and then the integration with the accelerometer.

5.2.1 Heading Information (Magnetometer)

In the header file of the controller that manages the location inputs from the GPS was added as well the functions for receiving and parsing the information from the magnetometer. We declared the methods `startHeadingUpdate` and `stopHeadingUpdate` from the `CLLocationManager [APP10]` class which return void and declared the method

```
locationManager:(CLLocationManager*)aManager didUpdateHeading(CLHeading *) aHeading
```

which returns void, from the `CLLocationManagerDelegate` class in order to tell the delegate that the location manager received new updated heading information.

The `startHeadingUpdate` method is called when the use of the magnetometer is specifically activated in the settings of the application or when the routing mode is changed to pedestrian mode. In this method, first we detect if there is a heading available with the `headingAvailable` method. If there is a heading available then we define a filter to prevent having an application that is too sensible to change in order to limit the inputs. This also contributes to decrease the battery consumption but since we have a navigation application, the value for the filter cannot be very high or else if we changed direction slightly the error would be too great after a short period of time. Acceptable values for these types of applications are between 2 to 5 degrees Celsius. We defined 2 degrees meaning that for each 2 degree changes the system would get a heading update and communicate it to the delegate of the location manager. Finally, if there is a valid location manager object we start the generation of updates that report the user's current heading, [`locationManager startUpdatingHeading`]. If there is no heading available, it means that the device does not have a magnetometer and so we show a message warning the user that there is no heading and the device does not have a digital compass.

```
- (void)startHeadingUpdate{
    if (locationManager.headingAvailable) {
        locationManager.headingFilter = 2;
        if(locationManager != nil)
            [locationManager startUpdatingHeading];
    }
    else {
        UIAlertView *noCompassAlert = [[UIAlertView alloc] initWithTitle:@"No
Compass!" message:@"This device does not have the ability to measure magnetic fields."
delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
        [noCompassAlert show];
        [noCompassAlert release];
    }
}
```

After starting the generation of heading updates, for each update, the `didUpdateHeading` method is called. For this method we need to have location data like latitude, longitude, speed, among others. So, if we don't have this information this method is useless. After saving the

Sensor Enhanced Navigation (Magnetometer and Accelerometer)

value for the heading accuracy from the update received, we check the heading accuracy value. If its greater or equal to zero it means the value is acceptable and so we set the heading accuracy value and the heading value in the location data class with the values received from the heading update.

```
-(void) locationManager:(CLLocationManager *)aManager didUpdateHeading:(CLHeading *)aHeading{

    if ( locationData == NULL ) {
        // without location data this method is useless
        return;
    }

    headingAccuracy = [aHeading headingAccuracy];

    if(headingAccuracy >= 0){
        locationData->setHeadingAccuracy(headingAccuracy);
        heading = [aHeading trueHeading];
        UIDeviceOrientation orientation = [[UIDevice currentDevice] orientation];

        locationData->setHeading(heading);
    }
    else
    {
        locationData->setHeadingAccuracy(
LocationData::kLocationDataInvalidHeadingAccuracy );
        locationData->setHeading(LocationData::kLocationDataInvalidHeading);
    }
    locationData->commit();          // notify the system
}
```

As for the stopHeadingUpdate, this method tells the location manager object to stop the generation of heading updates. If the location manager object is not null and therefore is valid, then we stop the generation of the heading updates, [locationManager stopUpdatingHeading].

```
-(void)stopHeadingUpdate{
    if(locationManager != nil)
        [locationManager stopUpdatingHeading];
}
```

Also, when the location manager fails with an error or the user denies the use of the Location Services (needed to determine the user's position), the heading update is stopped as well.

5.2.2 Acceleration information (Accelerometer)

Just like for the heading information, we have a `startAccelerometerUpdate` and a `stopAccelerometerUpdate` which do the same things explained before for the acceleration information [APP10]. The only differences are that for the acceleration, in the `startAccelerometerUpdate` we initialize the accelerometer

```
UIAccelerometer *accelerometer = [UIAccelerometer sharedAccelerometer];
```

and then we configure it by defining the update interval and then the delegate object.

```
accelerometer.delegate = nil;
accelerometer.updateInterval = 1.0 / kUpdateFrequency;
accelerometer.delegate = self;
```

Finally, we set the flag in the location data class at true, meaning that the generation of acceleration updates has been activated.

To stop the accelerometer updates in the `stopAccelerometerUpdate` method, we just define set the accelerometer delegate at nil and set the `accelerationReceived` flag at false.

```
accelerometer.delegate = nil;
accelerationReceived = NO;
```

As for the `didAccelerate` method we need to implement a high pass filter to filter the gravitational acceleration and use only the inputs from moving the device. In the definition of the high pass filter that was used, it was defined a minimum update frequency for the acceleration values received from the accelerometer, minimum step and value for the noise attenuation.

```
#define kAccelerometerMinStep          0.02
#define kAccelerometerNoiseAttenuation 3.0
#define kUpdateFrequency               40.0
```

The HPF formula used was as follows:

For each of the values received from each axis:

```
double dt = [[UIAccelerometer sharedAccelerometer] updateInterval];
```

We need to conserve the value of the update interval for the accelerometer in order to know the maximum amount of how many values per second I can expect and to use this value in the definition of the `filterConstant` variable. Before we can calculate the value for this variable we need to calculate the value for the time constant τ which is inversely proportional to the cutoff frequency of the HPF. The τ constant is defined as the RC variable.

```
double RC = 1.0 / 5.0;
```

Sensor Enhanced Navigation (Magnetometer and Accelerometer)

Now, since we have τ and the time interval dt we can finally calculate the filterConstant.

```
double filterConstant = RC / (dt + RC);
```

```
double d = Clamp(fabs(Norm(x, y, z) - Norm(aAcceleration.x, aAcceleration.y, aAcceleration.z)) / kAccelerometerMinStep - 1.0, 0.0, 1.0);
```

After that, we calculate the value for the weight of the filter which will determine how much importance the contribution of some forces will be given, like for example how much will the force of gravity contribute to the value that will be considered as the acceleration received from a determined axis.

```
filterConstant = (1.0 - d) * filterConstant / kAccelerometerNoiseAttenuation + d * filterConstant;
```

Last but not least, we calculate the actual value for the acceleration for each of the axis. We perform the difference between the value received of unfiltered acceleration and the weighed value for the unfiltered acceleration and add the contribution of the previous value of acceleration in order to achieve a more realistic value for the acceleration and avoid spikes in acceleration and unstable measurements of acceleration.

```
valueAxis = valueReceived - (valueReceived * filterConstant) + (previousValueReceived * (1.0 - filterConstant))
```

Finally, we save the current values for each axis in the lastX, lastY and lastZ variables.

5.3 Chapter summary

The implementation and use of the information provided by these sensors brings advantages to the navigation by offering an alternative solution to some of the existing problems. By using the magnetometer, the user has the possibility of knowing which direction he is orientated to. Also, this information can be used to determine the orientation of the pedestrian where no GPS signal fix is available such as inside buildings or on urban canyons. By using the accelerometer to determine if the user is moving or not and defining a fixed speed for the pedestrian when moving, it is possible to simulate an animation where the position of the user is extrapolated based on simple physics formula for determining the traveled distance. This can be done when the GPS signal is lost and correct the position of the user when the signal is regained. By combining the information provided by the two sensors it is possible as well to use this technique to perform indoor navigation on buildings where the map is available.

However, these sensors have some disadvantages as well. The magnetometer needs to be recalibrated when there is strong magnetic interference close by and the accelerometers integrated in most of the recent mobile devices still tend to be a little unstable and fragile and on several tested iPhones it was determined that the calibration of the sensors tends to be slightly different on some devices, especially when the device suffers some accident like a blow for example.

Chapter 6

Conclusions

6.1 Final Remarks

After analyzing all the main parts of what an information system for pedestrian navigation should have, we can conclude that there is still a lot of work to be done. Mobile devices are evolving at a steady and relatively rapid pace which leads us to conclude that we should expect evolution of the devices' processing capabilities and memory storage capacity in the next few years. With this evolution, new solutions will come to improve the experience of the pedestrian user in navigation that have better accuracy and more functionalities that will make navigation almost seamlessly both when outdoor and indoor as well.

With this dissertation we have a basis of a pedestrian navigation system with public transport information. In order to be able to run on different mobile devices, the architecture had to be based on solid concepts that rely on the abstraction level. The Data Abstraction Layer is what makes possible for the system to be implemented on several platforms.

There is a data model which may not be as complete as others at the basic level of the map composition (nodes, edges, POIs, Areas, etc...) but offers a model for extending the core part of the map through the addition of a public transport information package and a pedestrian additional coverage information package. It also explained how new information could be added in order to have costs to trips included in the information system. This information regarding the costs of trips is not so relevant to add as is for example the schedule information for the various lines of the various transport systems as shown in the study by Grotenhuis et al. [GRO07]. People are generally more concerned about time, delays and route information and advices rather than actual costs. Costs would be more important mainly for people who are strangers to a particular location like for example tourists. For tourists it would compensate to have this cost information because it would allow them to save money in transportation expenses around the city. Nonetheless, it would be even better for these types of users to have a service which would provide them with a package of scenic routes that would pass along the main and most important tourist attractions in that location, for example. In order to have the cost information about the several transport types, associated zones which derive themselves from area polygons of the basic core data model information. Most public transport tariffs are classified using zones and so that makes this information necessary.

Another information package that is proposed in the data model is the indoor navigation package which takes advantage of some of the definitions added in the pedestrian additional

Conclusions

coverage package, such as the virtual connection and access points. By introducing this information in the data model and with the growing interior cartography of big public buildings such as shopping malls and major stations, indoor navigation is one of the most probable next steps in making both indoor and outdoor navigation almost seamless.

Since the most recent smartphones are equipped with integrated sensors and that is becoming a widespread practice, it makes sense to use these sensors in order to achieve new solutions for some of the traditional problems presented by pedestrian navigation. The implementation presented on the iPhone 3GS is such an example and it was shown how the information provided by the integrated sensors allowed navigation when the GPS signal was lost or even inside buildings. A HPF was implemented in order to filter out the influence of gravity on the accelerometer having a possible way to determine when the user is moving and when he is not and through the magnetometer it was possible to determine the orientation of the user at all times. However, these sensors are not perfect and sometimes their behaviour is not the expected. So, although being a possible solution the quality of the integrated sensors still has to improve in order to be considered a truly viable alternative solution.

As for the adaptivity to the user, it was defined in the architecture where the profiling component would fit and a possible way to implement this Profile Configurator. Also, information that is important to regard for these purposes is considered in the data model.

6.2 Future Work

Pedestrian navigation is an area with a very broad scope. There are many parts to be considered in order to have an information system for pedestrian users of public transport that covers all the needs of all the types of users. For example, the requirements for blind people were not considered in this work.

Since the main goals of this dissertation were the specification of the solution, an architecture to support it, a spatiotemporal data model and a multimodal routing algorithm that would use the information contained in the data model, some other parts were neglected. One such part is the navigation aids. Despite the voice navigation package being present in the architecture, the best way to present the navigation instructions to the user was not determined or how they could be adapted to each user through the profiler component presented on the architecture.

Also, another possible functionality that could be integrated in this information system but was left out was the inclusion of augmented reality, both for directing the user to its goal in a route by pointing out the orientation to the location of the destination and having the sign marking it grow bigger just as we get closer to it; or for displaying additional information to the user about the POIs for example.

The input from the accelerometer could be improved as well by implementing a Kalman Filter instead of a HPF [ROJ]. Kalman Filter are more complex than HPF and have been proven to work well with accelerometer like for example in the Wii Remote [RAS07].

Regarding indoor navigation, the information added to the data model allows navigation on the interior of buildings but how this information is represented in the graphical interface while navigating was not studied.

As for the routing algorithm, there is still a need to define an algorithm and its heuristics which would make use of all the information in the data model. One example is how to differentiate between routing through the outside of a building or taking the shortcut through the inside because the mechanics of shortcuts on graphs were also not studied.

Finally, in order to verify all the mentioned above, a prototype implementation is needed.

References

- [BAR06] Barbeau, S., Labrador, M., Winters, P., Pérez, R., Georggi, N., A General Architecture in Support of Interactive, Multimedia, Location-Based Mobile Applications, IEEE Communications Magazine, November 2006.
- [CHE08] Chen, K., Dow, C., Guan, S., 2008, NimbleTransit: Public Transportation Transit Planning Using Semantic Service Composition Schemes, Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008.
- [CHR75] Chriqui, C., Robillard, P., 1975. Common bus lines. *Transportation Science* 9, 115-121.
- [FRA07] Frank, A., 2007, Wayfinding for Public Transportation Users as Navigation in a Product of Graphs, *Vermessung & Geoinformation* 2/2007, Pages 195-200.
- [GRO07] Grotenhuis, J., Wiegmans, B., Rietveld, P., 2007, The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings, *Transport Policy* 14, Pages 27-38.
- [HUA07] Huang, R., 2007, A schedule-based pathfinding algorithm for transit networks using Pattern First Search. *GeoInformatica*, 11, Pages 269–285.
- [HUA08] Huang, R., Peng, Z.-R., 2008, A spatiotemporal data model for dynamic transit networks, *International Journal of Geographical Information Science*, 22: 5, Pages 527-545.
- [PEN97] Peng, Z.R., Nebert, D., 1997. An internet-based GIS data access system. *Journal of Urban and Regional Information Systems* 9 (1), 20-30.
- [PEN00] Peng, Z.-R., Huang, R., 2000, Design and development of interactive trip planning for web-based transit information systems, *Transportation Research Part C* 8, Pages 409-425.
- [PON06] Pontikakis, E. (2006): *Wayfinding in GIS: Formalization of Basic Needs of a Passenger When Using Public Transportation*. Vienna, Technical University Vienna. Doctor of Technical Science.

REFERENCES

- [SMI04] Smith, J., Mackaness, W., Kealy, A., Williamson, I., 2004, Spatial Data Infrastructure Requirements for Mobile Location Based Journey Planning, *Transactions in GIS*, 8 (1), Pages 23-44.
- [SPE94] Spear, B., 1994. GIS and spatial data needs for urban transportation applications. In: David Moyer, D., Ries, T. (Eds.), *Proceedings of the 1994 Geographic Information Systems for Transportation (GIS-T) Symposium*, Norfolk, Virginia, pp. 31-41.
- [SPI89] Spiess, H., Florian, M., 1989. Optimal strategies: a new assignment model for transit networks. *Transportation Research B* 23, 83-102.
- [ZIL07] Ziliaskopoulos, A., Aifandopolou, G., Chrisohoou, E., 2007, A multi-objective optimum path algorithm for passenger pre- trip planning in multimodal transportation networks. *Journal of The Transportation Research Board*.
- [ZIP06] Zipf, A., Jöst, M., 2006, Implementing adaptive mobile GI services based on ontologies Examples from pedestrian navigation support, *Computers, Environment and Urban Systems*, Volume 30, November 2006, Pages 784-798.
- [ZOG08] Zografos, K., Androutsopoulos, K., 2008, Algorithms for Itinerary Planning in Multimodal Transportation Networks, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 1, March 2008.
- [RUL10] RuleML Homepage, 2010 – “RuleML, The Rule Markup Initiative”, <http://ruleml.org/> (last access on: 02/05/2010)
- [HEC03] Heckaman, D., 2003, Introducing “Situational Statements” as an integrating Data Structure for User Modeling, Context-Awareness and Resource-Adaptive Computing, In: *ABIS2003*, Karlsruhe, Germany (2003) 283–286
- [GAR93] Garlan, D., Shaw, M., 1993, An Introduction to Software Architecture, In V. Ambriola and G. Tortora (eds), *Advances in Software Engineering and Knowledge Engineering*, vol. 2, World Scientific Publishing Company, 1993, pp.1-39.
- [WES99] West, M., Fowler, J., 1999, Developing High Quality Data Models, The European Process Industries STEP Technical Liaison Executive (EPISTLE).
- [RAS07] Rasco, B., 2007, Where's the Wiimote? Using Kalman Filtering To Extract Accelerometer Data, http://www.gamasutra.com/view/feature/1494/wheres_the_wiimote_using_kalman_.php (last access on: 17/03/2010)
- [APP10] iPhone OS Reference Library , Core Location http://developer.apple.com/iphone/library/documentation/CoreLocation/Reference/CLHeading_Class/Reference/Reference.html (last access on: 07/05/2010)

REFERENCES

- [RET04] Retscher, G., Thienelt, M., 2004, NAVIO – A Navigation and Guidance Service for Pedestrians, Presented at GNSS 2004, The 2004 International Symposium on GNSS/GPS
- [MAY03] May, A., Ross, T., Bayer, S., Tarkiainen, M., 2003, Pedestrian navigation aids: information requirements and design implications, *Pers Ubiquit Comput* 7:331
- [DEN01] Denis, M., Michon, P.-E., 2001, When and Why are Visual Landmarks Used in Giving Directions. In *Spatial Information Theory: Foundation of GI Science. Lecture Notes in Comp. Science*, D. R. Montello, Ed. Berlin: Springer 2001
- [ELI04] Elias, B., Brenner, C., Automatic Generation and Application of Landmarks in Navigation Data Sets, In Fisher, P., ed.: *Developments in Spatial Data Handling*. Springer (2004) 469-480
- [REH05] Rehrl, K., Goll, N., Leitinger, S., Bruntsch, S.: Combined indoor/outdoor Smartphone navigation for public transport travellers. *Location Based Services & Telecartography-Proceedings of the Symposium* (2005)
- [DZI06] Dziekan, K., Kottenhoff, K., 2007, Dynamic at-stop real-time information displays for public transport: effects on customers, *Transportation Research Part A* 41 (2007) 489–501
- [BAR08] Sean J. Barbeau, Miguel A. Labrador, Alfredo Perez, Philip Winters, Nevine Georggi, David Aguilar, Rafael Perez. “Dynamic Management of Real-Time Location Data on GPS-enabled Mobile Phones,” *UBICOMM 2008 – The Second International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies*, Valencia, Spain, September 29 – October 4, 2008.
- [ROJ] Rojas, Raul: The Kalman Filter. <http://www.inf.fu-berlin.de/lehre/WS04/Robotik/kalman.pdf>, (last access on:10/04/2010)
- [GOI07] Goix, L. et al., 2007, “Situation Inference for Mobile Users: A Rule Based Approach”, In *Mobile Data Management, 2007 International Conference on*. pp. 299-303
- [ARC10] ArcGisNetwokModel, <http://edndoc.esri.com/arcobjects/8.3/TechnicalDocuments/Network/ArcGISNetworkModel/ArcGISNetwork.htm> (last access on: 12/06/2010)

Appendix A

Requirements Specification for Public Transport Users

A.1 Introduction

In order to achieve an information system that complies with the user's requirements and therefore providing a better using experience for the user, first it is important to know what those requirements are. In this chapter it is presented functional requirements through the use of use cases, and non-functional requirements for public transport users.

A.2 Requirements

In this section it is presented the requirements specification for an information system for public transport users through use case models. The use cases were divided into Services, Routing & Navigating, Search and Configuration but first we present below an overview of the use case model.

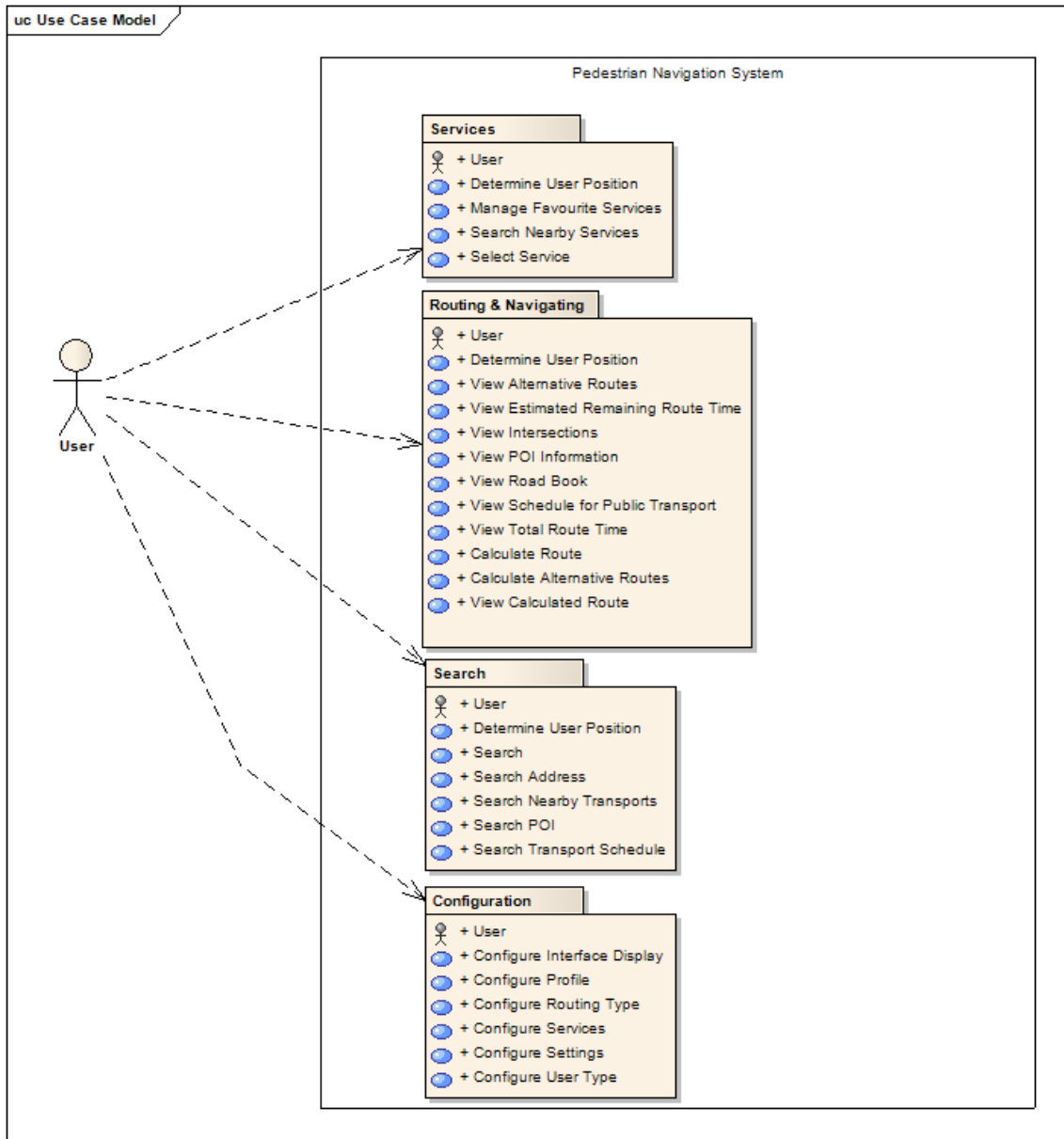


Figure A.1: Use case model for the information system

Table A.1: Services Use Case Package Description

Description	Services
Main Scenario	
The User can access the functionalities provided by the Services Package.	
Actors	User

Table A.2: Routing & Navigating Use Case Package Description

Description	Routing & Navigating
Main Scenario	
The User can access the functionalities provided by the Routing & Navigating Package.	
Actors	User

Table A.3: Search Use Case Package Description

Description	Search
Main Scenario	
The User can access the functionalities provided by the Search Package.	
Actors	User

Table A.4: Configuration Use Case Package Description

Description	Configuration
Main Scenario	
The User can access the functionalities provided by the Configuration Package.	
Actors	User

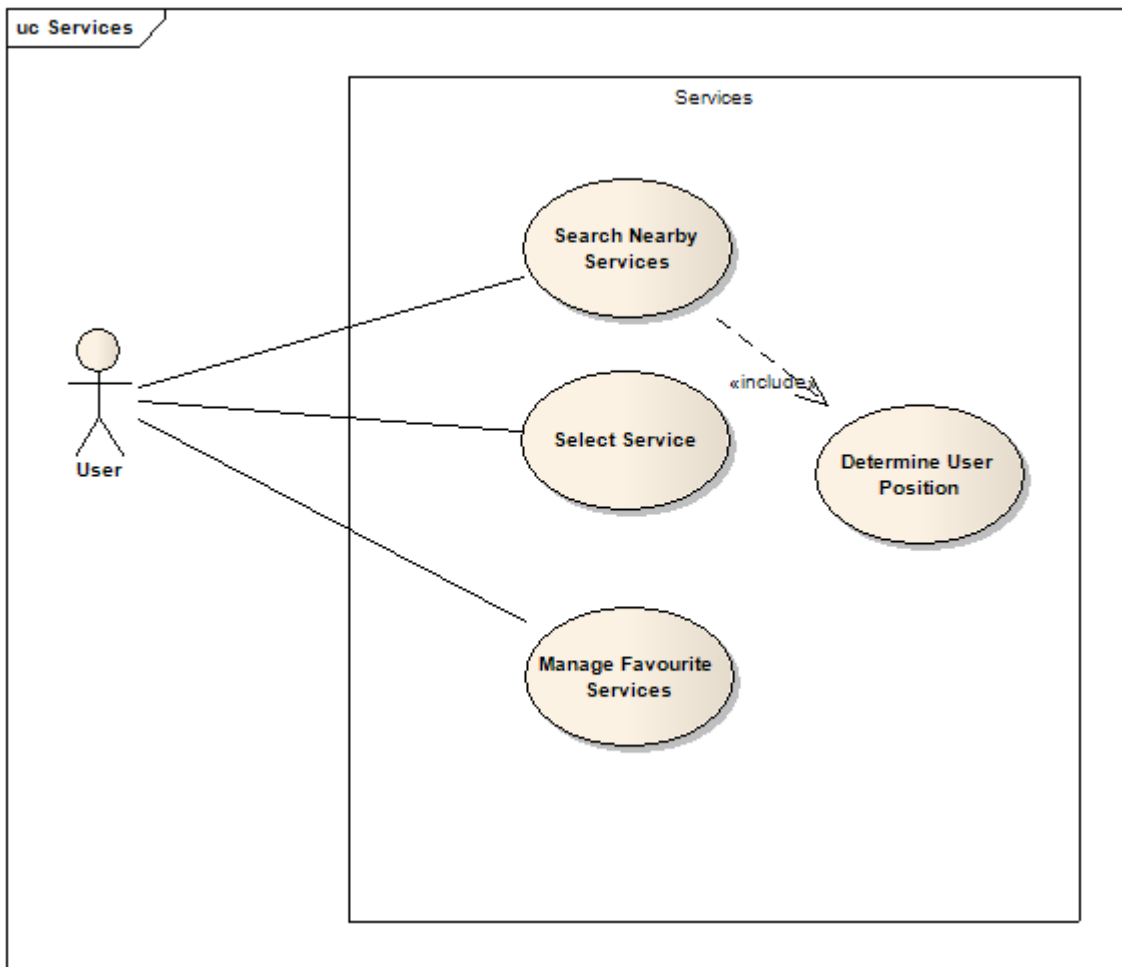


Figure A.2: Services Use Case Model

Table A.5: Search Nearby Services Use Case

Description	Search Nearby Services
Main Scenario	
The User accesses the option to search nearby services and the list of detected services in the vicinity is shown as well as the User’s previously added favourite services.	
Actors	User
Pre-conditions	N/A
Post-conditions	A list of services available is shown.
Assumptions	The User must possess an Internet connection in his/her mobile device.
References	N/A

Table A.6: Select Service Use Case

Description	Select Service
Main Scenario	
The User selects a service from the list of available services. After the selection the User can add it to his list of services (if not previously added) or use it.	
Actors	User
Pre-conditions	The list of available services is shown.
Post-conditions	The user can use the service or add it to the list of favourites.
Assumptions	The User must possess an Internet connection in his/her mobile device.
References	N/A

Table A.7: Manage Favourite Services Use Case

Description	Manage Favourite Services
Main Scenario	
The User can manage the services available in his list of favourite services by either removing or editing them, or add new available services to the list of favourites.	
Actors	User
Pre-conditions	N/A
Post-conditions	All changes must be visible.
Assumptions	The User must possess an Internet connection in his/her mobile device and in case of removing or editing services they must previously exist.
References	N/A

Table A.8: Service's Determine User Position Use Case

Description	Determine User Position
Main Scenario	
The system can determine the User's position via GPS or by extrapolation via the use of sensors of the device.	
Actors	User
Pre-conditions	N/A
Post-conditions	The system determines and shows the User's current position.
Assumptions	The system has a valid GPS fix or sensors that allow for a relatively accurate position determination and they are correctly calibrated.
References	N/A

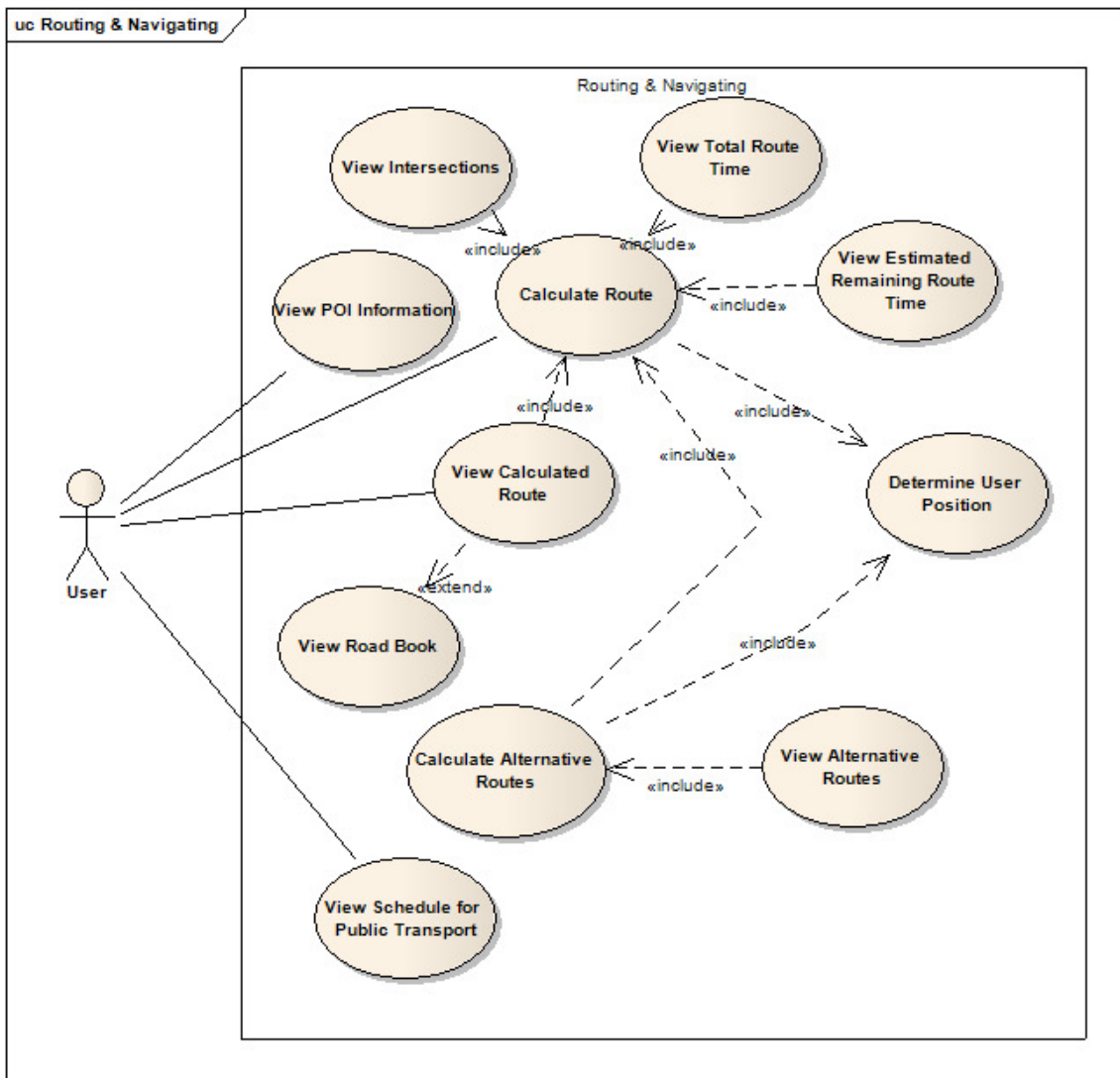


Figure A.3: Routing & Navigating Use Case Model

Table A.9: Calculate Route Use Case

Description	Calculate Route
Main Scenario	
The User gives the system the destination point and for the route and the system utilizes the User's current position for the point of origin for the route. For the calculation of the route, the system also uses the current configuration for the setting and the User's profile to calculate the most suitable route with those characteristics.	
Actors	User
Pre-conditions	The system has at least valid coordinates for the destination point.
Post-conditions	A route that suits the requirements is presented.
Assumptions	N/A
References	N/A

Table A.10: View Calculated Route Use Case

Description	View Calculated Route
Main Scenario	
The User is presented with the calculated route that best suits the objectives and configurations of the User.	
Actors	User
Pre-conditions	There is a calculated route.
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.11: View Schedule for Public Transport Use Case

Description	View Schedule for Public Transport
Main Scenario	
The User can select and view the schedule for a particular public transport line.	
Actors	User
Pre-conditions	The User has selected a particular public transport line and that line exists.
Post-conditions	The schedule for the selected public transport line is shown to the User.
Assumptions	There is information regarding the schedule or at least the frequency of the public transport in the data model.
References	N/A

Table A.12: View Intersections Use Case

Description	View Intersections
Main Scenario	
The User can view the intersections between public transport and/or public transport lines that he/she has to travel through or if there are no intersections in the route the system shows a message informing the user.	
Actors	User
Pre-conditions	The system has previously calculated a route.
Post-conditions	The list with the intersections is shown to the User.
Assumptions	N/A
References	N/A

Table A.13: View Total Route Time Use Case

Description	View Total Route Time
Main Scenario	
The system presents the user when previously solicited in the configurations the total route time for the current calculated route.	
Actors	User
Pre-conditions	There is a previously calculated route.
Post-conditions	The system shows the total route time for the calculated route.
Assumptions	N/A
References	N/A

Table A.14: View Estimated Remaining Route Time Use Case

Description	View Estimated Remaining Route Time
Main Scenario	
The system presents the user when previously solicited in the configurations the estimated remaining route time for the current calculated route.	
Actors	User
Pre-conditions	There is a previously calculated route and the system knows the user's position.
Post-conditions	The estimated remaining route time is presented to the user.
Assumptions	The system knows the user's current position.
References	N/A

Table A.15: View Road Book Use Case

Description	View Road Book
Main Scenario	
The User can view all the instructions that he has to follow to make it to the end of the calculated route.	
Actors	User
Pre-conditions	There is a previously calculated route.
Post-conditions	The list with all the directions is shown to the User.
Assumptions	N/A
References	N/A

Table A.16: Calculate Alternative Routes Use Case

Description	Calculate Alternative Routes
Main Scenario	
The system, by order of the user, calculates other routes that fit the objectives and characteristics of the user's configuration and profile but which are not considered the best solution that fits the user's needs.	
Actors	User
Pre-conditions	The best route was previously calculated.
Post-conditions	An alternative route is presented to the user.
Assumptions	N/A
References	N/A

Table A.17: View Alternative Routes Use Case

Description	View Alternative Routes
Main Scenario	
The system presents the user with a list of alternative routes that suit the objectives and profile of the user's configuration.	
Actors	User
Pre-conditions	The system has calculated the alternative routes.
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.18: Routing & Navigating's Determine User Position Use Case

Description	Determine User Position
Main Scenario	
The system can determine the User's position via GPS or by extrapolation via the use of sensors of the device.	
Actors	User
Pre-conditions	N/A
Post-conditions	The system determines and shows the User's current position.
Assumptions	The system has a valid GPS fix or sensors that allow for a relatively accurate position determination and they are correctly calibrated.
References	N/A

Table A.19: View POI Information Use Case

Description	View POI Information
Main Scenario	
The user can access and view all the information regarding a selected Point Of Interest.	
Actors	User
Pre-conditions	The user has selected a valid POI.
Post-conditions	The user is presented with the information available regarding the POI.
Assumptions	N/A
References	N/A

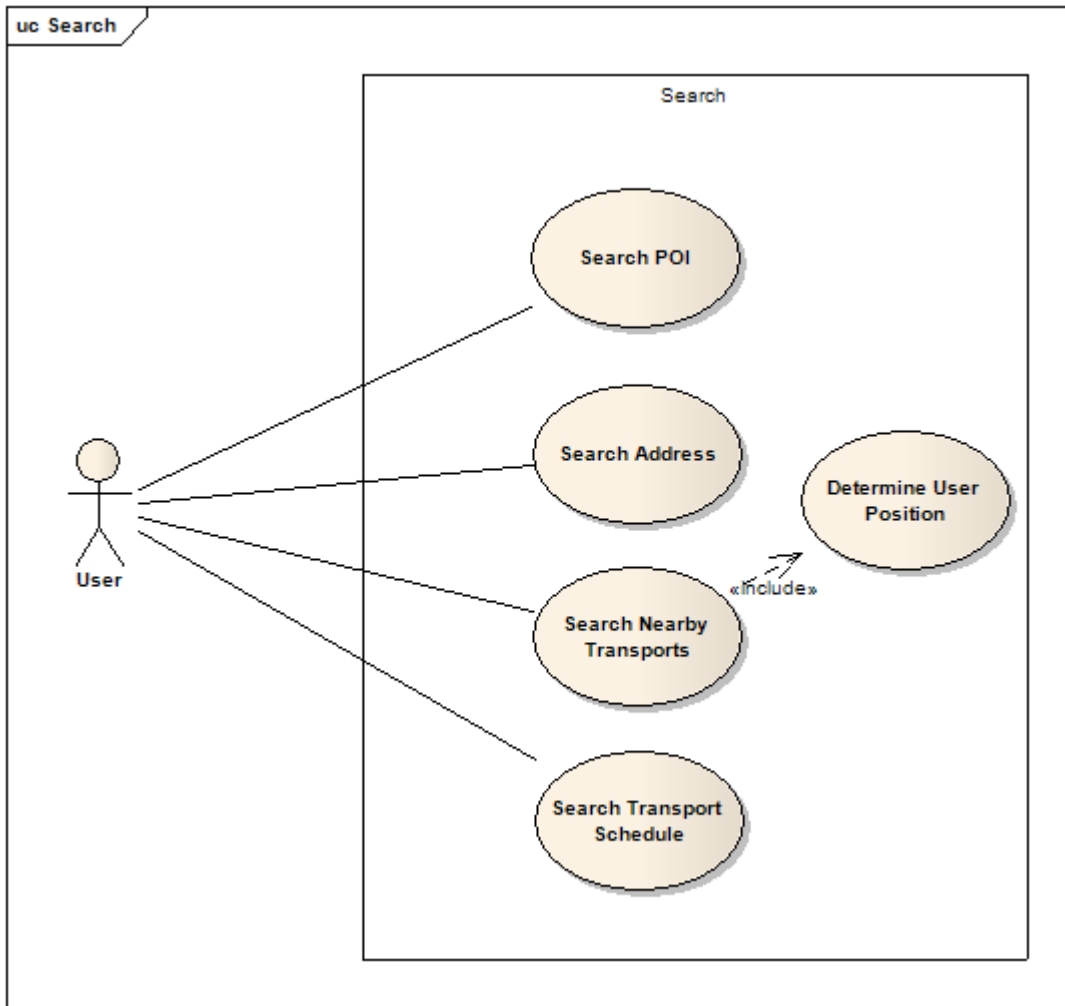


Figure A.4: Search Use Case Model

Table A.20: Search POI Use Case

Description	Search POI
Main Scenario	
The user can search for a particular Point of Interest by first giving the system the name of the location and then using the POI's name, address or zip code.	
Actors	User
Pre-conditions	N/A
Post-conditions	The Information regarding the POI is presented to the user.
Assumptions	N/A
References	N/A

Requirements Specification for Public Transport Users

Table A.21: Search Address Use Case

Description	Search Address
Main Scenario	
The user can search for a particular address.	
Actors	User
Pre-conditions	N/A
Post-conditions	The information regarding the address being searched is presented.
Assumptions	The address exists.
References	N/A

Table A.22: Search Nearby Transports Use Case

Description	Search Nearby Transports
Main Scenario	
The system searches for nearby public transport stops and/or stations.	
Actors	User
Pre-conditions	N/A
Post-conditions	A list with all public transport
Assumptions	The system has determined the user's current position.
References	N/A

Table A.23: Search's Determine User's Position Use Case

Description	Determine User's Position
Main Scenario	
The system can determine the User's position via GPS or by extrapolation via the use of sensors of the device.	
Actors	User
Pre-conditions	N/A
Post-conditions	The system determines and shows the User's current position.
Assumptions	The system has a valid GPS fix or sensors that allow for a relatively accurate position determination and they are correctly calibrated.
References	N/A

Table A.24: Search Transport Schedule Use Case

Description	Search Transport Schedule
Main Scenario	
The user can search for a particular public transport's schedule or frequency of departure. The user first has to select the city and then filter the search by giving the system the public transport type and finally the public transport line.	
Actors	User
Pre-conditions	The user gives the system the city, type of transport and line.
Post-conditions	The user is presented with the schedule for the public transport in question.
Assumptions	There is valid information regarding the public transport in the data model.
References	N/A

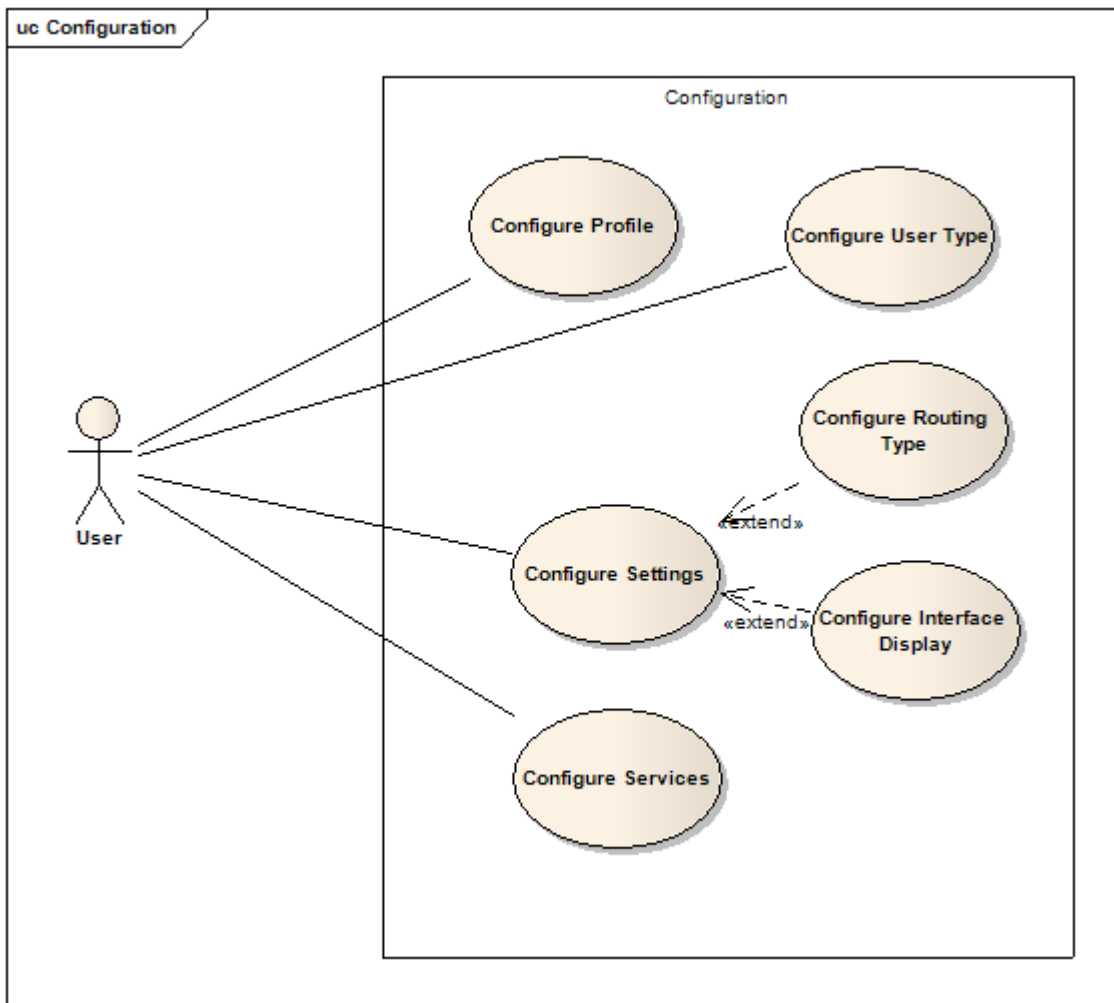


Figure A.5: Configuration Use Case Model

Table A.25: Configure Profile Use Case

Description	Configure Profile
Main Scenario	
The user can configure his/her profile options, making the system more adequate for the type of objectives the user has.	
Actors	User
Pre-conditions	N/A
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.26: Configure User Type Use Case

Description	Configure User Type
Main Scenario	
The user can configure some particular options of the user type defined. For example, if paths with no stairs are to be considered in the route planning process.	
Actors	User
Pre-conditions	N/A
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.27: Configure Settings Use Case

Description	Configure Settings
Main Scenario	
The user can configure the system settings making it a more personalized experience. Among the options are the metric system being used, the routing type how to present information in the interface display, among others.	
Actors	User
Pre-conditions	N/A
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.28: Configure Routing Type Use Case

Description	Configure Routing Type
Main Scenario	
The user defines the type of routing that he wishes the system to use. Along with the profile configuration settings, these options will influence the route calculation.	
Actors	User
Pre-conditions	N/A
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.29: Configure Interface Display Use Case

Description	Configure Interface Display
Main Scenario	
The user has access to a number of options that will give him/her a more personalized feel of the system use by adjusting the way information is presented in the graphic interface display.	
Actors	User
Pre-conditions	N/A
Post-conditions	N/A
Assumptions	N/A
References	N/A

Table A.30: Configure Services Use Case

Description	Configure Services
Main Scenario	
The user can configure the options relative to the services module of the system.	
Actors	User
Pre-conditions	N/A
Post-conditions	N/A
Assumptions	N/A
References	N/A

Along with the requirements presented before in the form of use cases, there are other requirements (the informal requirements) that an information system for public transport users should have in order to raise the quality of the system such as the following:

Scalability

The system should be designed having in mind its future growth, whether it is in the amount of information of the data model, functionalities or services. Due to the system's modularity and low coupling, the system is better prepared to be scalable.

Modularity

The system should be designed the most modularly as possible in order to facilitate eventual changes in individual components without causing a big impact in the overall functioning of the system.

Low coupling

In order to achieve a higher maintainability and a well structured design with a lower interdependency between modules, the system should present a low coupling. By being loosely coupled, the system will present the advantages of not forcing a ripple-effect in other modules when a change in a particular module needs to be implemented; the assembly of modules might require less effort and/ or time due to the low dependency between modules; and it makes reusability and/ or testing easier, again due to the fact of decreased module interdependency.

Usability

The system should be compatible with most of the recent smartphones and mobile devices.

Performance

All the modules in the system should function with an acceptable speed of performance, in concurrency with the technologies used.

Portability

The system should be as portable as possible in order to be executed correctly in most mobile devices as possible.

A.3 Chapter Summary

In order to have a starting ground for the specification of a solution for public transport users of pedestrian navigation systems, the specification of the requirements for these information systems was needed. So, using use cases, in this chapter it was presented the various requirements for the users of these systems when regarding functionalities and the non-functional requirements that the users desire in these types of systems.