A1-D7

NL Navigation Commands from Indoor WLAN fingerprinting position data

Project title:	Reasoning on the Web with Rules and Semantics
Project acronym:	REWERSE
Project number:	IST-2004-506779
Project instrument:	EU FP6 Network of Excellence (NoE)
Project thematic priority:	Priority 2: Information Society Technologies (IST)
Document type:	D (deliverable)
Nature of document:	R (report)
Dissemination level:	PU (public)
Document number:	IST506779/Munich/A1-D7/D/PU/a1
Responsible editors:	E.-P. Stoffel, B. Lorenz, H. J. Ohlbach, M. Rosner
Reviewers:	Georg Treu
Contributing participants:	Munich, Malta
Contributing workpackages:	A1
Contractual date of deliverable:	30 August 2006
Actual submission date:	30 September 2006

Abstract

Keyword List
semantic web, WLAN fingerprinting, NLG, indoor, navigation, route descriptions

Project co-funded by the European Commission and the Swiss Federal Office for Education and Science within the Sixth Framework Programme.

NL Navigation Commands from Indoor WLAN fingerprinting position data

Hans Jürgen Ohlbach, Mike Rosner, Bernhard Lorenz, Edgar-Philipp Stoffel

Department of Computer Science, University of Munich
E-mail: {ohlbach,lorenz, stoffel}@pms.ifi.lmu.de
Department of Computer Science & AI, University of Malta
E-mail: mike.rosner@um.edu.mt

30 September 2006

Abstract

Keyword List
semantic web, WLAN fingerprinting, NLG, indoor, navigation, route descriptions

Contents

This deliverable is based on the following works:

- Edgar Stoffel's diploma thesis on hybrid geospatial graph models [8, 4];
- Andreas Heindel's diploma thesis on the use of indoor positioning data for wayfinding [2];
- Thomas Rickinger's diploma thesis on WLAN fingerprinting [6]. This work was supervised by Axel Küpper and Georg Treu from Prof. Linnhoff-Popien's team at the Institute for Informatics, LMU Munich. We had a close cooperation with this team by jointly supervising the work of Andreas Heindel and Thomas Rickinger.
- Doreen Mizzi's master thesis about a Mobile Navigation Assistance System Using Natural Language Generation [5]. She developed a natural language guiding system for the main library of the University of Malta. She was supervised by Mike Rosner from the Malta team.

1 Introduction

The notion "semantic" in "semantic web" has a broad range of interpretations and realisations. It ranges from adding meta data to data, over developing ontologies for special domains, to developing detailed world models as the basis for information processing. In WG A1 we are in particular concerned with developing geospatial world models and testing them in applications.

In this deliverable, we describe one of the applications of geospatial world models and geospatial information processing: guiding persons through indoor environments. This scenario envisions that eventually the semantic web and the area of ubiquitous computing will merge into something which could be called the "ubiquitous semantic web". It is an instance of a much more general scenario: guiding persons from any location A to any location B on this planet using any means of transportation.

The architecture of a system for this purpose could look like in Fig. 1.

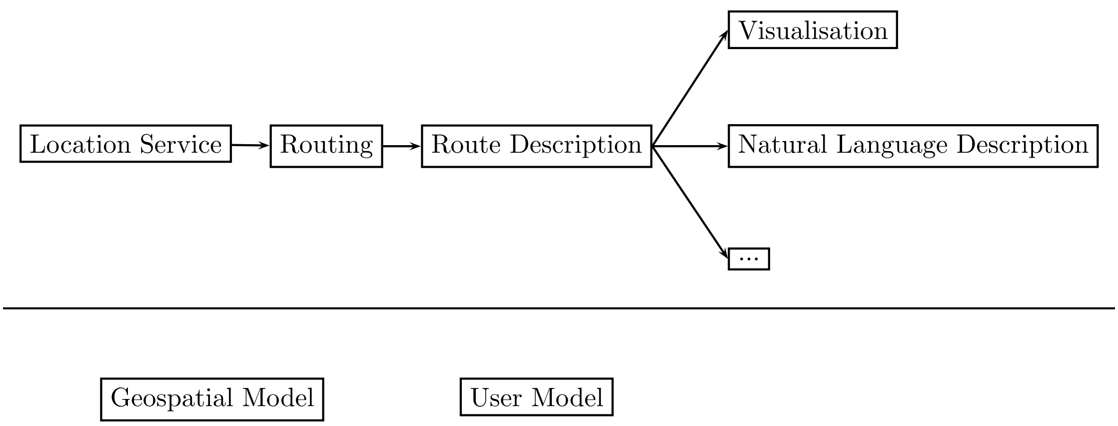


Figure 1: Guidance System

The basic components are the geospatial model of the environment, together with a user model. The geospatial model contains a representation of the geometry and topology of the environment. It is basically a graph with some extra geometric information (coordinates etc.). The user model describes the abilities and preferences of the user. For example, if he sits in a wheelchair, he must use a lift instead of stairs.

The *Location Service* is a tool for measuring the current geographic position of the user. This can be a GPS tool or, as we shall see in this deliverable, a device using WLAN fingerprinting. The geographic position must be correlated with the geospatial model. For example, if the geospatial model represents rooms and corridors in a building, it must be possible to associate the geographic position with the corresponding location in the building.

The *Routing* algorithm determines the path from the current position of the user to the position of the target. It is usually a shortest path algorithm. If there is a user model, the abilities and preferences of the user must be taken into account by the cost function of the shortest path algorithm. For example, if the user is pushing a pram, the cost of using stairs must be very high.

The routing algorithm generates a sequence of nodes and edges in a graph. This sequence must be turned into a description of the way to be followed. For example, if the route contains a link which represents a lift between the first and third floor, the route description must contain something like 'request the lift', 'enter the lift', 'press the button for the third floor', 'exit the lift'. That means, the sequence of nodes and edges must be turned into a sequence of actions. The description of the actions may refer to landmarks along the path, e.g. 'pass the statue of Max Planck'. We are currently developing an XML-based description language for such paths. It represents sequences of actions in an abstract and symbolic way such that various output formats can be generated from it.

Finally the route description is turned into different output formats. It can be visualised, or turned into natural language sentences etc.

For the time being, the various components are in different states of development. A few prototypic systems have been implemented which demonstrate various aspects of the system. In this deliverable we describe the ideas and the current state of the developments. We mainly concentrate on indoor scenarios because this is our main testbed. We are, however, by no means restricted to indoor environments only.

The user modelling part is not yet in a state we can present here. Therefore it is omitted.

1.1 Overview of Our System Components

In the course of the previously mentioned diploma and master theses, several prototypic components of a guiding system were realised in WG A1.

The very first part was the work of Mizzi [5]. Basically, it is a system for generating natural language (NL) navigation commands from geometrical descriptions of an indoor environment (the main library of the University of Malta). In a graphical editor, geometries resembling to floor plans can be constructed and stored. The focus of her work is on the linguistic aspects. The system is further described in Section 4. From there on, there were some points of extensions.

Firstly, the system provided an interface to a positioning system which had to be connected to it. The positioning system was realised in the diploma theses of Rickinger [6] who developed a WLAN fingerprinting technology (further details in Sect. 3.2).

Secondly, the original system was extended with respect to several important aspects in the work of Heindel [2]. In cooperation with Rickinger, the positioning system was connected to the NLG system. Beforehand, path planning was possible, among others, only for one floor. This entailed an extension of both the routing algorithm and the editor for the virtual environment. Beyond, route sketches as symbolic descriptions of navigation instructions have been added as well. The analysis of Heindel, as a by-product, revealed some shortcomings in the implementation of the geometric algorithms. As a consequence, in some cases the generated instructions were incorrect.

This last observation has led to us to reconsidering the model underneath the indoor environment. It is an essential component, since both positioning system and NLG are linked to it. To this end we developed a hybrid graph model which combines concrete and symbolic data about indoor environments. A very first version, the *TransRoute system* has been implemented by Stoffel in his diploma thesis [8]. The model is currently being revised and extended [4]. We start with the description of this model. An overview over the other works is given in the subsequent sections. Details can be found in the corresponding diploma and master theses.

2 A Hybrid Geospatial Graph Model

Paths from some position A to some other position B are usually computed by a shortest path algorithm in a graph. These algorithms only need the nodes and edges of a graph, together with a cost function. It turns out that much more complicated than computing a shortest path is the problem of turning the shortest path into a human understandable description. A lot of additional information is needed to generate useful path descriptions. Therefore, we developed a hybrid graph model which combines structural and geometric information to serve both, the shortest path algorithms and the generation of descriptions.

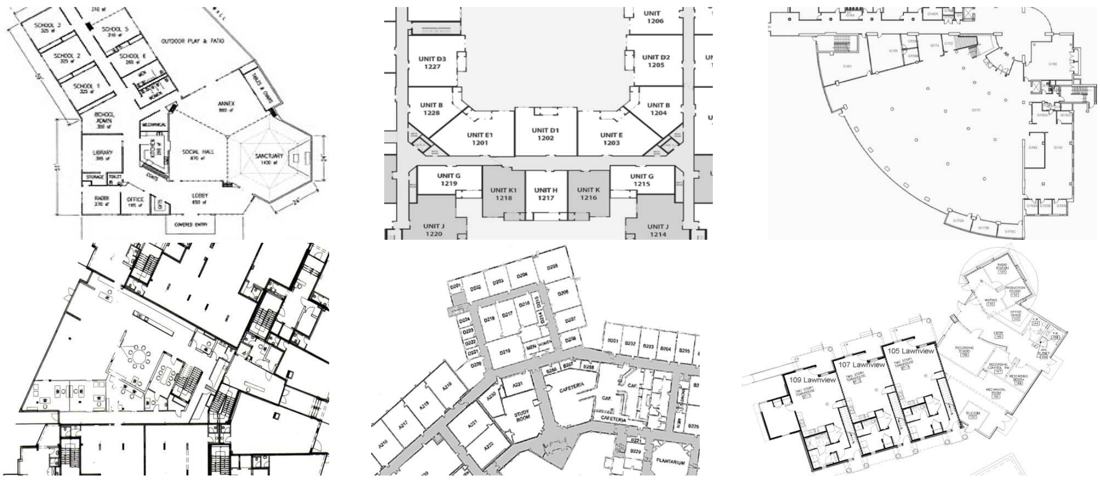


Figure 2: Various Geometries Occurring in Floor Plans

The main characteristics of the graph model are:

- The two-dimensional areas in buildings are partitioned into cells, and these cells are represented as nodes in the graph (Section 2.1). Doors and other passways which represent possibilities for persons to move from one cell to another are represented as edges;

- In order to facilitate hierarchical planning, there are different levels of abstraction in the graph (Section 2.2). For example, a storey in a building may be represented as a graph at a certain level, this entire graph being just a node in a graph at a higher level which stands for the whole building. The edges in the abstract graph connect the different storeys;

- The nodes and edges of the graph are labelled with hybrid information to support wayfinding as well as the generation of a human-understandable description of a path. Hence, we distinguish different types of nodes (Section 2.3). For example, rooms and corridors are both represented as nodes, but with different labels. As we shall see, it is quite useful to maintain a list of doors and windows in a room, all sorted by their angle against a fixed point of reference (Section 2.4). Corridors, on the other hand, are essentially one-dimensional structures for which it is useful to maintain the sequence of doors at the left hand side and the sequence of doors at the right hand side (Section 2.5).

The indoor model is described in more detail in the subsequent sections. However, we want to emphasise that the model is deliberately kept flexible. The node and the edge types as well as their labelling can be extended when it turns out that this is suitable for future applications.

2.1 Cell Decomposition

For buildings with simple rooms and corridors, that is to say rather small rooms (unlike, for instance, an entrance hall where hundreds of people fit in) and narrow enough corridors (not stretching over several parts of a building), there is a direct one-to-one mapping to a graph structure. Rooms and corridors are represented as nodes, and the passways between them as edges. In Fig. 3, where an extract from a blueprint of a university building is shown, such a graph structure is laid over the floor plan.[1] Two rooms which are connected by two or more doors have two or more edges between the corresponding nodes (like the entrance hall and the main corridor in Fig. 3).

However, strictly pursuing this naïve approach becomes difficult for larger buildings with large areas of open space, as for instance an airport. Following Bittner [1] we divide the free space C_{free} in this case into non-overlapping, disjoint cells C_r such that $C_{free} = \bigcup_r C_r \land \forall i \neq j : C_i \cap C_j = \emptyset$. Adjacent cells are connected by a link. The main corridor in Fig. 3 is actually split into several cells due to its length. Otherwise, impractical route descriptions like *"turn left to the main corridor and take the 32nd door on the right"* may result.

The sheer size of a room may be a reason to decompose it into cells. Other reasons have to do with concavity of rooms, or with the functionality of certain areas in a larger open space. For example, an airport lounge may feature waiting areas, meeting points, areas in front of the different counters and security checks, passport control, etc. All of them serve a different purpose, and this must be represented in the graph.

Unfortunately, there is no obvious way to fully automate the cell decomposition. It has to be designed very carefully, taking into account the purpose of the different cells.

2.2 Hierarchical Graphs

If you are at the first floor of a large building, and you ask someone how to get to a particular room, the explanation may well start with *"Go to the third floor ..."*. What is behind this is a two-level (or, in general, multi-level) hierarchical model of the building. An example is depicted in Fig. 4. The upper hierarchy level consists of the storeys, and the lower level models the

[1] The stairs to the other two storeys were omitted for keeping the example simple.

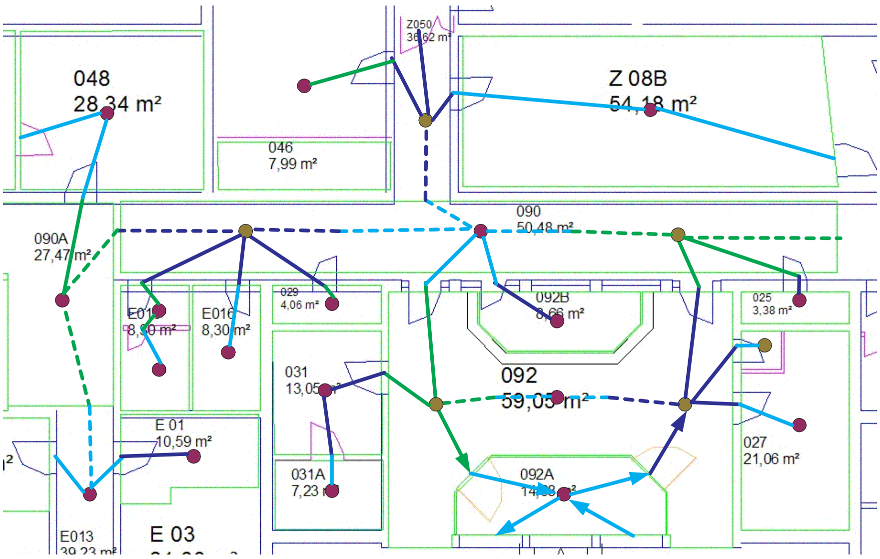


Figure 3: Floor Plan Overlaid with Cell Centres and a Path System

topology of each storey. In addition, the hierarchy shown in Fig. 4 also has an intermediate level which consists of wings. Navigation between different storeys usually consists of the steps *"go to the lift (staircase etc.)"*, *"go to the target storey"*, *"navigate the target storey"*. This is a typical case of hierarchical planning as it has been investigated in Artificial Intelligence for decades [7].

Our graph model supports hierarchical planning by providing hierarchical graphs. Each graph has a level (in the hierarchy) and an identifier. Graphs at higher levels contain nodes which can be labelled with the identifiers of graphs at lower levels. But this is not enough. There must also be a possibility to access graphs at higher levels from nodes of graphs at lower levels. This is done by classifying certain nodes of graphs at level n as "interface nodes" to graphs at level $n+1$. Physically, these interface nodes may represent access points to staircases, lift doors, etc. (see Fig. 4).

The primary use for the graph hierarchy is of course the representation of different storeys in a building. Other use cases may necessitate the representation of different wings in a building (as in Fig. 4). Wings and storeys yield a hierarchy of three levels. If it makes sense to subdivide wings further, one may have four or more levels (see Fig. 5). On the other hand, there may also be further levels above the level of storeys. If we want to represent not only a single building, but, say, the whole campus of a university with many buildings, each building would be a node in a graph one level above the level of storeys.

A further use of hierarchical graphs can be the representation of areas which are contained within each other. As an example, consider the vegetable area in a hypermarket. The vegetable area may be subdivided into the area with the salad, the cucumbers, the carrots, etc. In the hierarchical graph model, we would have a node for the vegetable area at some level n, and this node refers to the graph of the salad, cucumber etc. areas at level $n-1$.

The edges in the graph at the 'building level' represent walkways or streets. In the simplest case, such an edge contains solely the information that it is *possible* to get to another building.

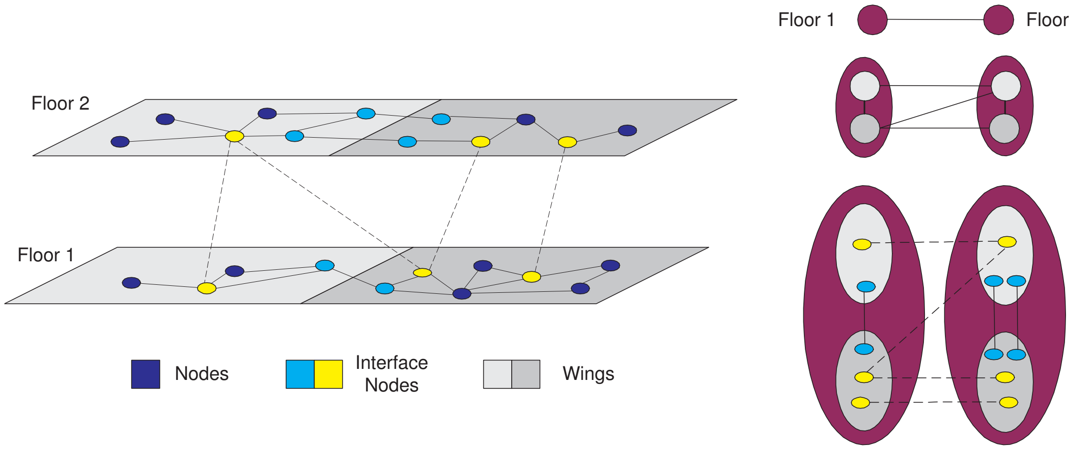


Figure 4: Hierarchical Graph

If we want more detail on *how* to get to this building, we must link the edge with another graph which describes the walkways and the road network. Therefore not only nodes of a graph at level $n+1$ can represent graphs at level n, but also edges at level $n+1$ can represent graphs at level n. The only difference is that an edge at level $n+1$ must correspond to a graph at level n with two interface nodes, one for each end of the edge.

Granularity \ **Model Element**	**Graph**	**Vertex**	**Edge**
Level 4 (coarsest)	City	Building	StreetNetwork
Level 3	Building	Storey, *Staircase*, *Elevator*	
Level 2	Storey	Wing, Room, Corridor	Door, Window, Ladder, Ramp, Stairs
Level 1	Wing	Room, Corridor	
Level 0 (finest)	Room, Corridor	PartOfRoom, PartOfCorridor	

Figure 5: Relations between Hierarchy Levels and Graph Elements

2.3 Node and Edge Types

Wayfinding by means of shortest path algorithms requires no more than a graph and a cost function. A simple cost function measures the geometrical distance between two places. More sophisticated cost functions can, for example, distinguish between lifts and staircases by making the staircases more "expensive". A minimum of semantic information is sufficient for this purpose. It turns out that the problem of wayfinding is considerably easier than the problem of describing an indoor path in a human-understandable manner. Humans use a combination of mostly qualitative information ("use the door *at the end* of the corridor") with little quantitative information ("take the *second* door to your left") for describing routes. Landmarks, which are very important in outdoor scenarios ("after passing by the church"), however, seem to be less important for indoor scenarios, although Mizzi [5] uses them extensively to characterise both long paths ("pass a row of shelves on your left") and points of arrival ("here should be a computer on your right") more naturally.

In order to support the generation of descriptions of a path through a building, we need to enrich the graphs with a lot more semantic information. Therefore it is necessary to classify indoor areas and to attach further class-dependent information to the nodes and edges. In this paper we illustrate the node types with two examples, namely 'rooms' and 'corridors'. Other types could be 'waiting area', 'meeting point' etc. In hierarchic graphs with many levels, node types like 'wing', 'storey', 'building' etc. are needed.

The node and edge types correspond directly to an ontology of building components. At present it is, however, not yet clear whether it is possible to describe the ontology in a formal description language like the Web Ontology Language, in short OWL[2], and to automatically incorporate the OWL concepts into the graph data structures. If this were indeed possible, it would make the graph framework much more elegant and flexible.

2.4 Rooms

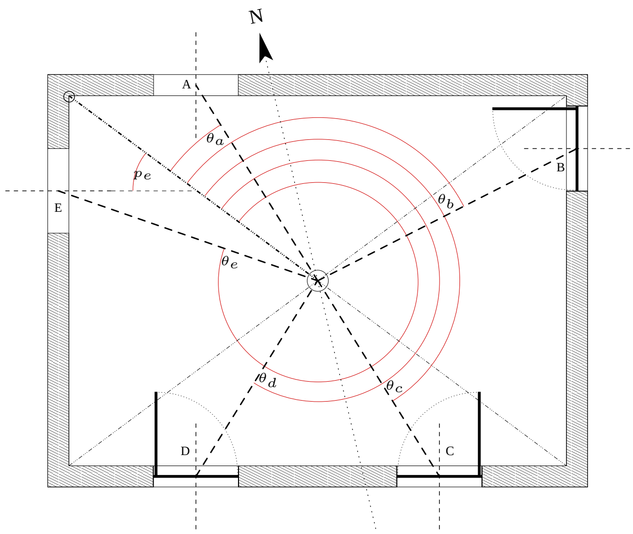


Figure 6: Hybrid Model of a Room

Rooms which are not further decomposed into cells are represented by a single node. Each door is represented by an edge leading to the neighbouring rooms. This is not enough information for generating instructions like *"take the second door on your left"* with the optional clarification

[2]http://www.w3.org/TR/owl-features/

"[the door] directly opposite the window to your right". In the event of further information being available, one could of course add the coordinates of the corners of a room and those of all doors and windows. It turns out that for generating instructions like the ones above, it is sufficient to have a less complex data structure, such as a list of angles between the doors (or windows, respectively) and a reference line which goes from the centre of gravity of the room to a fixed reference point at the wall (we use the most north-western corner). An example is depicted in Fig. 6.

A path crossing the room by entering through door B and leaving through door D may, for example, be described by the statement *"take the second door on your left"*. The information *"second door on your left"* can be computed as follows: the trajectory from B to the centre divides the room into *left* and *right*. Doors C and D are to the left and windows A and E are to the right. This can be derived from the angular distribution of the doors and windows. Thus, D is 'to your left'. The fact that D is the second door on your left can simply be obtained, by counting the number of doors in clockwise direction from B to D.

The further clarification *"[the door] directly opposite the window to your right"* can only be generated when the angular orientation of the walls is also stored. Together with the orientation of the doors and windows one can find out whether there exists another door or window which is situated opposite to door D.

The methods described above implicitly assume that the person entering at door B is looking towards the middle of the room. For this case it is sufficient to store a single angular distribution at the room node. If, however, the person looks straight forward when he enters the door, his notion of left and right may be different. Window E would now be to the left instead of right, for example. To account for this, one must compute the angular distribution for each door separately and store it at the corresponding end of the edge that represents the door. The line of reference for the angles crosses the middle of the door and is perpendicular to the door.

For many notions there are phrases in the human language which describe these notions with varying degree of precision. For example, there exist several degrees of *opposite*, such as *somewhat opposite*, *fairly opposite*, and *directly opposite*. A possible mathematical representation of these fuzzy notions are *fuzzy sets*, in our case fuzzy angular distributions (see Fig. 7)[3]: This has the advantage that deviations from an angle, like for the notion of being opposite, can still be regarded as being opposite, but only to a certain degree (determined by the membership, a fuzzy value between 0 and 1). The choice whether to use, for example, *somewhat opposite* or *fairly opposite* can be done by evaluating the corresponding fuzzy values on the distribution. If, say, the fuzzy value for the angle has been evaluated to 0.6, it would qualify as *somewhat opposite* whereas 0.95 would be considered as *directly opposite*. It is practical to use several intervals with decreasing threshold values for the various levels of 'opposing'.

2.5 Corridors

There are in fact two ways for modelling corridors. The first method is to decompose a corridor into cells such that each entrance to the corridor can be associated with a representative cell. Adjacent cells are represented by edges between the corresponding nodes. The main corridor in Fig. 3 can modelled this way, leading to a representation of seven cells for the seven adjoining doors. This representation is completely sufficient for solving wayfinding problems. It is,

[3] Depending on the application, core angles θ_c (fuzzy value of 1) and support angles θ_s (fuzzy value > 0) can vary.

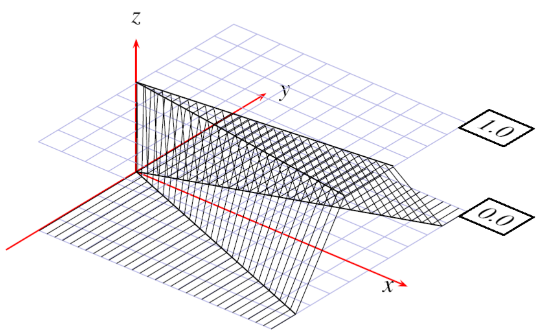


Figure 7: Fuzzy Angular Distribution ($\theta_c > 0$, $\theta_s > \theta_c$)

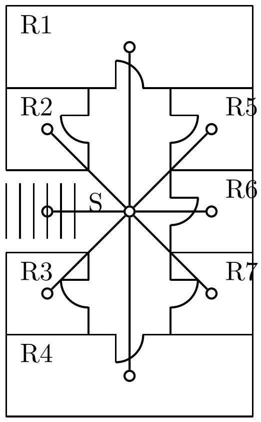
however, very cumbersome to generate a statement like *"take the third door to the right"* and from a practical point of view, it is certainly not the most elegant and compact[4] data structure. The second way is illustrated in Fig. 8. The whole corridor is represented by a single node. However, this node actually stands for a directed linear structure leading from the front to the end, with openings both on its left hand side and its right hand side.[5] It does not affect the general notion of linearity whether the corridor is distorted, since the notions of 'left' and 'right' are relative and thus change accordingly. The node must have labels which represent the entrances at the left side of the corridor, the entrances at the right hand side of the corridor, and the entrances at both ends of the corridor. The list of edges must reflect the real sequence of doors, stairways etc. It must keep the distances between two subsequent elements as well, or the offset from the front. Using these lists, it is easy to reconstruct from a particular door and a particular orientation an instruction like *"go to the second door on your left"*. The main corridor in Fig. 3 could be partitioned into several of such sequences.
A prototype version of this graph model has been implemented by Stoffel in the *TransRoute* system [8].

3 Location Service via WLAN Fingerprinting

3.1 Overview of Indoor Positioning

One possibility for representing a position is by means of quantitative, measurable coordinates of a certain coordinate system [3]: *WGS-84* is a prominent example for such a coordinate system which is wide-spread. WGS-84 defines coordinates in accordance with a reference geoid. The Global Positioning System (*GPS*), among others, gives back *WGS-84* coordinates. Internally, GPS uses Earth Centered Earth Fixed (*ECEF*) coordinates. ECEF is a Cartesian coordinate system originating at the centre of gravity of the Earth, and defined by the prime meridian (Greenwich) and the equatorial line. ECEF coordinates are then transformed into the more practical geodetic WGS-84 coordinates. However, for a small scale scenario like navigating

[4] in terms of storage
[5] in a way, the node is a dual to the linear structure



Labelling of the corridor node:

Left:	R3,S,R2
Right:	R7,R6,R
Front:	R4
End:	R1

Figure 8: Corridor

inside a building, it is rather a burden than a benefit. In this case, it seems that symbolic coordinates are more appropriate.

Traditionally, positioning methods are used to determine the position of an object by calculations involving other objects of reference (e.g. by triangulation or lateration). But this is by no means the only way to determine the position of an object. There are also other methods which use identifiers (like RFID) and sensors, or analyse images by their similarity (scene analysis). Of course, combinations of different methods are also possible. One can basically categorise positioning methods as follows [6]:

- Geometrical: Angulation and Lateration
- Scene Analysis: Image Analysis and Fingerprinting
- Proximity Sensing: Sensor-based

A positioning system consists of a certain positioning method together with a physical infrastructure. Indoor environments necessitate special positioning systems. Systems like *GPS*, which rely on satellites in orbit, cannot yet be used for positioning in buildings. However, the heterogeneity of indoor positioning methods is remarkable. A comparison involving some of the numerous different indoor positioning systems used in practice can be seen in table 1 below. The type of signal, for instance, ranges from ultrasound to radio and infrared. Each method has its characteristic strengths and limitations, and certainly none is perfect. There is a trade-off between cost, precision and applicability to be weighed up.

As explained above, all of these indoor positioning systems require a very specific technical infrastructure (mostly in the building). Considerable cost and time, however, have to be devoted for setting up such an infrastructure (or complete network). Therefore, solutions which make use of existing infrastructures like a Wireless Local Area Network[6] (*WLAN*) are of particular interest.

[6] Defined in the IEEE Standard Family 802.11, most commonly denoting the standards 802.11b and recently 802.11g with a transfer rate of 11 Mbit/sec. resp. 54 Mbit/sec.

Name	Basic Method	Type of Signal	Measurement	Type of Network
Active Bat	Lateration	Ultrasound	Time	418 MHz radio
RADAR	Fingerprinting	Radio	RSS	WLAN
Ekahau	Fingerprinting	Radio	RSS	WLAN
RFID	Proximity	Radio	ID	
Indoor GPS	Lateration	Radio	Time	
Active Badge	Proximity	Infrared	ID	
WIPS	Proximity	Infrared	ID	WLAN

Table 1: Different Indoor Positioning Systems Compared

3.2 WLAN Fingerprinting

WLAN fingerprinting is one specific form of scene analysis (see table 1). In this case, 'scene' is not to be taken literally in the sense of classical image analysis. Rather, a characteristic electromagnetic profile of signals forms a scene in this context: The radio signals used for wireless transmission create a very specific signal pattern (depicted in Fig. 9). This phenomenon now can be exploited by WLAN fingerprinting.

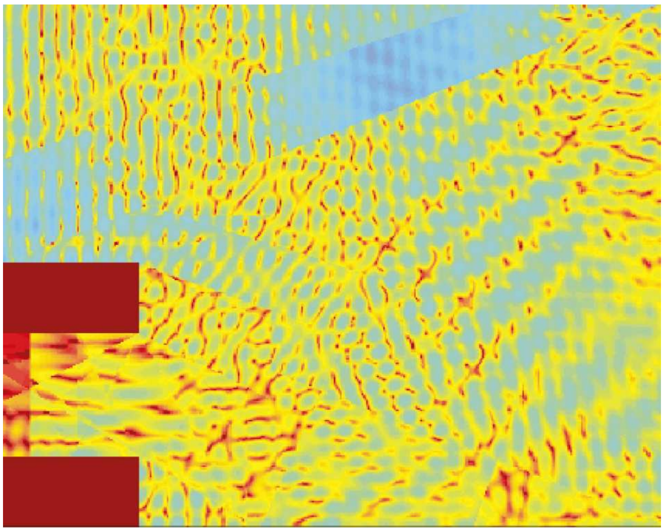


Figure 9: WLAN Fingerprint by Received Signal Strength [6]

Signals are measured in terms of Received Signal Strength (RSS) from an access point. Therefore, this technique resorts to matching patterns of measured signals with signals previously recorded and stored in a database (the fingerprints). This first step is done in the so-called offline-phase. Fingerprints are known signal patterns recorded at selected points of reference. Formally, they can be regarded as vectors which relate a reference position to several RSS, one for each base station. Practical experience has shown that several measurements of a RSS yield more accurate results.[7]

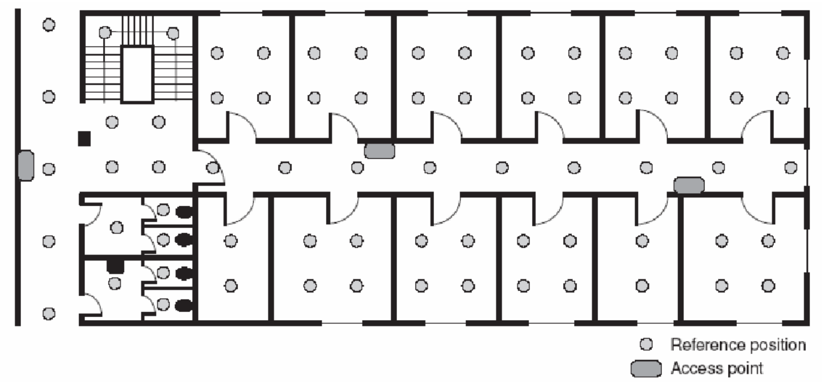


Figure 10: Radio map with several access points and reference points [3]

Several of these fingerprints are collected in a so-called *radio map* (see Fig. 10) in the preliminary offline-phase before the system is actually run (real-time phase). An obvious disadvantage is that whenever changes or events occur, the recorded signals have to be updated. Otherwise, wrong matchings could be the consequence. If many people move around in the building, the signal profile may be additionally falsified. A pragmatic, yet inelegant solution to this problem would be using different radio maps for different times of day, say one for rush hour and another one for less busy hours.

The actual positioning is done by measuring the signal strengths of the WLAN access points which can be received, and comparing the signal strengths with the pre-recorded map. A single measurement usually gives only approximate results, and even worse, it can yield several equally likely positions, even at different floors of the building. To further improve the reliability of the results, one can use history-based user tracking. That means the system records the user's movements and exploits the assumption that the user cannot instantly jump over large distances (Startrek's beamer technology is not yet available). When a new position is to be measured, the system compares the most likely positions with the movement of the user and orders them according to some plausibility heuristics. In the implementation of Rickinger, the reliability of the generated positions improved from around 75% without user tracking to about 95% with user tracking. The precision of the generated coordinates was about 1 meter. These results are very competitive with regard to other indoor positioning systems. In particular, the system works without any extra cost in almost every building where WLAN is installed.

The results of the measurement are packed into the NMEA 0183 protocol and sent via bluetooth to other applications. This way the interface to the system is the same as the interface to GPS systems. Therefore it is no problem to combine the indoor positioning system with a GPS system for outdoor positioning in a way which is completely transparent for the application systems.

Furthermore, for indoor scenarios it is no problem to get the correlation between geographic

[7]This is due to the erratic diffusion of radio signals. Mathematical models can only provide a simplification of this phenomenon.

coordinates and room identifiers. During the phase where the radio-map is recorded one can easily fill up a lookup table with the correlation between the coordinates and the room identifiers. This way one gets in addition to the coordinates also room identifiers as a result of the location service.

4 NL navigation commands

In this section we give a very brief overview on Mizzi's work on NL generated instructions for navigating in a library. Since NL generation is not a main focus of WG A1, we keep it short. The details can be found in [5].

4.1 Landmarks

For the domain of indoor environments, not only the spatial structure itself has to be represented, but also the categories of solid, physical objects which occupy these spaces (like tables, chairs, or persons as well). They serve, for example, as landmarks which can be used in instructions like 'go around the table'. Mizzi describes in her thesis a system [5] in which all domain objects are modelled in an object-oriented programming language (*C#*) by a class hierarchy with inheritance. This conceptual hierarchy distinguishes several types of space:

- **standalone object:** no other domain objects are contained in it. From the viewpoint of a hierarchical tree for spatial containment, standalone objects are leaves, and thus at the bottom level.
- **container object:** can contain other domain objects in its spatial area example: table with objects on top of it, or below it
- **space object:** Spatial area or region which is actually a *void* area delimited by *boundaries*, e.g. a room with walls and a ceiling. Being subclass of the container object, a space object may, of course, be filled with other domain objects.
- **tangible container object:** Subclass of container object, too. In contrast to a void spatial area, it is a solid physical object. Examples are a shelf, a seat or a table.

4.2 The System Components

Mizzi's system consists of the following components:

The Path Planner This components constructs a search graph by decomposing rooms into rectangular cells which can be empty, partially empty of completely filled by obstacles. The construction is supported by a quadtree. The neighbourhood relation of the decomposed cells yields a search graph which is the basis for the A* algorithm to find a shortest path in the graph.

The Text Planner This component turns the route-based path plan into a series of messages. Each message can be considered as a concept that can be realised linguistically. For example, the concept of going from one end to another in a spatial area can be linguistically realised as "moving along a spatial area".

The approach is based on a repertoire of top-level generic message types originally derived by inspecting a manually collected *corpus*. More specific messages inherit from these abstract types, using specialisation on the inheritance hierarchy. Several samples of input and output are gathered, which are afterwards analysed. At this, good quality of the samples is of vital importance. The analysis revealed five basic types of messages, found at the top level of the inheritance hierarchy:

- **Path Message** for describing movement within a spatial area
- **PathDoor Message** for describing movement through a door
- **Point Message** for describing the current position of an object
- **Direction Message** for describing a change of direction
- **Observe Message** for describing a certain object in the current surroundings. This message can have an affirmative character, ensuring the the user that he still is on the right path, or else it may help to find the next important decision point. Either way, the object must be salient as it serves for human orientation (hence a landmark).

The Linguistic Realiser This is the last part in the pipelined architecture. It takes a message in the text plan and linguistically realises each message against a grammar. In addition it performs some form of sentence aggregation and referring expression generation.

4.3 Example of Generated Instructions

The following example gives an impression of the performance of the system.

> "Move up the main stairs and reach the door in front of you. Walk through the door. There should be an EU information computer on your left. Move across the corridor and reach the aluminium door on your right. Move through the aluminium door. Go straight ahead, along the main corridor. There should be a row of computers on your right. Pass by them until you get to the end of that corridor. Go through the aluminium door. Move to your left. Walk across the main door, towards the information desk. Reach it. Go straight ahead. Pass near the row of shelving sections on your left. The shelving section you want is the fifth one on your left. What you want is the first shelf on your left in the second row of shelves from top. There should be a green book. The book you want is the fifth black one to its right."

4.4 Extensions by A. Heindel

Heindel extended Mizzi's work by the following aspects [2]:

- The system was extended from one floor only to buildings with several floors, connected by stairs and lifts.
- The indoor positioning system of Rickinger was attached, such that the system can now generate path descriptions from the current position of the user to some destination.
- As an additional output medium, the natural language expressions were supported by pictograms on a display. The pictograms show mainly directions to walk. Without the NL sentences they would not be sufficient to guide a person. Nevertheless, they are easy to see and understand, and they help the user.

5 Summary

The prototypic system yields more or less satisfactory results in very restricted domains. For a more widely usable system, which combines indoor and outdoor scenarios, however, a number of improvements are necessary. Most of them are currently being developed.

- The location service must combine indoor and outdoor positioning. A number of research groups and commercial companies are working at this problem. This is therefore not an issue for WG A1.
- Our hybrid graph model needs further extensions. In particular for outdoor scenarios, we need a dynamic component which is able to model time–varying features, for example buses moving along streets. Static timetables of public transport systems might be sufficient to a certain degree, but for real world modelling, really dynamic features of the graphs are necessary. The next version of the TransRoute system should have such a feature.
- A further extension of the graph model is a combination of a user model and a modelling of the means of transportation. It is, of course, no problem to hard-code in Java that persons in wheelchairs cannot take the stairs, but use the lift instead. Much more flexible, however, would be to model both, users and means of transportation, in an ontology system, and to develop an algorithm which matches the user's needs with the capabilities of the means of transportation. This is also currently being investigated.
- In Mizzi's work, a route-based path plan was translated into a message. The message was then verbalised. We are currently refining the concept of message used there. The first step in translating a path to some output format is actually the translation of a sequence of nodes in the search graph into a sequence of *actions*. A sequence of actions could in principle be the basis for guiding a robot, as well as the basis for generating messages to be fed into either a NL generation system or a graphical visualisation system. The language PlanML, which is currently being developed in a diploma thesis, is an XML-based language for representing actions. PlanML is intended to be the exchange language for action plans, not arbitrary actions, but actions which movable objects need to perform to get from some place A to some place B with different means of transportation.

Acknowledgement

This research has been co-funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`).

References

[1] T. Bittner. The Qualitative Structure of Built Environments. In *Fundamenta Informaticae*, volume 46 (nr. 1–2), pages 97–128, 2001.

[2] A. Heindel. Nutzung von Indoor-Positionierungsdaten zur mobilen Wegeplanung. Diplomarbeit/diploma thesis, Dept. of Computer Science, University of Munich, March 2006.

[3] A. Küpper. *Location–based Services — Fundamentals and Operation.* John Wiley & Sons, Aug. 2005.

[4] B. Lorenz, H. J. Ohlbach, and E.-P. Stoffel. A Hybrid Spatial Model for Representing Indoor Environments. In *Proceedings of the 6th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2006) – forthcoming – December 4-5, 2006, Hong Kong, China*, Lecture Notes in Computer Science. Springer, 2006.

[5] D. Mizzi. A Mobile Navigation Assistance System Using Natural Language Generation. Diplomarbeit/diploma thesis, Dept. of Computer Science and AI, University of Malta, 2004.

[6] T. Rickinger. Eine Client/Server-Architektur zur endgerätebasierten Ortung auf Basis von WLAN Scene Analysis. Diplomarbeit/diploma thesis, Dept. of Computer Science, University of Munich, March 2006.

[7] E. D. Sacerdoti. Planning in a Hierarchy of Abstraction Spaces. In *Artificial Intelligence*, volume 5(2), pages 115–135, 1974.

[8] E.-P. Stoffel. A research framework for graph theory in routing applications. Diplomarbeit/diploma thesis, Institute of Computer Science, LMU, Munich, 2005.