Route Directions Generation using Visible Landmarks

Davide Russo
Department of Information Engineering, Computer Science and Mathematics,
University of L'Aquila,
Via Vetoio, Coppito, 67100
L'Aquila, Italy
davider87@gmail.com

Sisi Zlatanova
GIS Technology, Department OTB, Architecture and the Built Environment,
Delft University of Technology
Julianalaan 134, 2628 BL
Delft, The Netherlands
S.Zlatanova@tudelft.nl

Eliseo Clementini
Department of Industrial and Information Eng., and Economics,
University of L'Aquila,
Via G. Gronchi 18, 67100
L'Aquila, Italy
eliseo.clementini@univaq.it

ABSTRACT

The aim of this research is to investigate how to communicate route directions for wayfinding assistance in indoor environments including visible landmarks along the route. We propose an algorithm to automatically generate low level directions, as an XML file, that can be later translated in other languages, e.g., IndoorGML. The most suitable data model is a graph with openings (doors, windows, passages), features and concave corners as nodes, and edges based on geometrical visibility between them. For a given route, the proposed algorithm extracts all the surrounding visible nodes and groups them to simplify subsequent textual instructions. This process is then implemented in a software prototype, "IndoorNav", based on an Android device. It uses QRcode scanning for locating user position, calculates the best route to follow, generates low-level route directions, and translates them into textual instructions in the requested language; finally, it shows them to users.

Keywords

Indoor Navigation, Wayfinding, Landmarks, Location Based Systems, Android, Route directions.

1. INTRODUCTION

In order to reach places that satisfy our needs and wants, we need to know where to go and how to get there. According to Montello [23]: "Navigation" can be defined as the "*coordinated and goal-directed movement of one's self (one's body) through the environment*". As Montello proposed, it may be conceptualized as consisting of two components: 1) *wayfinding*, which involves localization of ourselves and the destination, and the choice of the right route to take; 2) *locomotion*, which refers to the movement of the body in the direction we intend, avoiding obstacles and barriers. In this paper, we deal with the wayfinding component of navigation and we are not considering locomotion issues. We will concentrate on providing directions to follow in wayfinding tasks. Indoor navigation is not a new concept, but in contrast to outdoor (or car) navigation, it has more and new challenges to deal with; Stoffel identified these key differences [31]:

- *Shape diversity*: the network structure of roads is regular and clearly defined, road segments are linear and at every junction of 3 or more segments there is a decision point; also, large metropolises show a grid pattern, especially modern districts. In contrast, a systematic treatment of indoor environments is difficult, architects have more freedom in designing a building, rooms can vary in size and shape depending on their function. Particularly large rooms can be unique for their shape and multitude of connections.
- *Degrees of freedom in movement*: vehicles are mostly bounded to lanes/rails, and drivers must respect driving rules: it is not allowed to turn just anywhere, reversing direction everywhere, stopping along the road everywhere, and so on. Pedestrian motion is less restricted than vehicles, since in large halls they can move freely. Fig. 1 shows the difference between outdoor vs indoor freedom movement.
- *Granularity*: the speed of vehicles and pedestrian is different while moving, and involves a different level of detail on the perspective of the surrounding space: it means that features of a building must be modelled at a higher granularity.
- *Network type*: road networks can be modelled by one-dimensional data structures, like graphs; on the other hand, it is much more difficult to extract path structures from building spaces, especially in large rooms in which we can only define a region of free moving. The transit in large spaces seems rather fuzzy!

Despite all above difficulties, every day we are able to find a route to our destination point, also in unfamiliar buildings. How do we give/understand/follow directions, sometimes in environments of very complex shape? The most common environmental features used in indoor route directions are *pathways*, *landmarks* and *choice points* [20]. We never use numerical references about distances or turning angles, since people prefer to use actions anchored to each multiple-choice

ISA'14, November 04-07 2014, Dallas/Fort Worth, TX, USA
Copyright 2014 ACM 978-1-4503-3137-1/14/11 ...$15.00
http://dx.doi.org/10.1145/2676528.2676530.

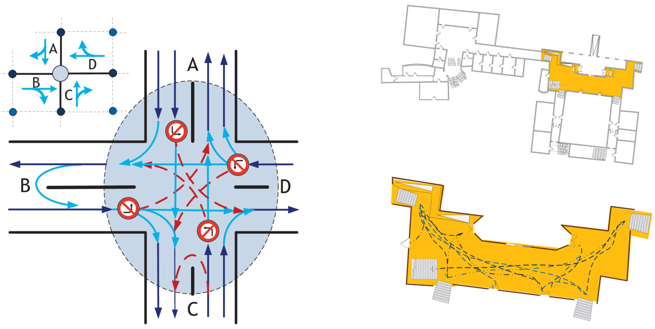


Figure 1: Motion in road networks vs pedestrians in indoor environments [31]

ambiguous location (decision points) and use *landmarks* to provide confirmation that the right track is still being followed (landmarks along route segments) [26].

Landmarks are defined as "prominent features in the environment that are unique or in contrast with their neighborhood" [28] and as "natural, built, or culturally shaped features that stand out from their environment" [13]. A landmark may also be described as an "environmental feature that can function as a point of reference that serves as sub-goal that keeps the traveler connected to both the point of origin and the destination along a specified path of movement" [20]. Landmarks lead to shorter learning times, better recall in route description tasks, and better response in wayfinding tasks [27]. Daniel and Denis [9] demonstrated that only about 15% of human direction elements are not related to landmarks. It is essential to integrate references to landmarks in wayfinding assistance to generate cognitively ergonomic route directions. Although they are widely used in human wayfinding and communication about routes, today's spatial information systems rarely make reference to them. The main reason is the lack of available data about landmarks or even agreed characteristics defining a landmark.

To address the challenges of indoor navigation, we take into account the following requirements:

1. *Open spaces modeling*: manage big open space through a network model that reflects the real human free movement with the minimum deviation to the real routes.
2. *Human orientation*: we consider an egocentric orientation system (not allocentric) that can compute user directions while moving and is capable to give instructions based on the individual current orientation.
3. *Landmarks usage*: as explained in previous sections, the usage of landmark is generally preferred in common human route directions. Our objective is to integrate guidance with landmarks.
4. *Hierarchical instruction*: multiple levels of detail in route instructions exploit the advantage of splitting long paths into smaller sequences giving higher level instructions and, if requested, other levels of detail.
5. *Directions generation adaptability*: we seek a dynamic route directions generation that reflects the changes of an editable data model. This requirement suits emergency situations or work in progress conditions, where route directions should take into account real-time changes, such as a corridor is unavailable due to a fire, a room is temporary closed for works, or an emergency exit should be taken. We would try to avoid a static route direction system, using recorded videos, or stored directions that are not flexible to daily scenarios.
6. *Internationalization*: a language independent route generation system can translate instructions in more than one language without changing the route generation process.

In the next section, we introduce the data model used for the route direction generation. In Section 3, we explain the route direction concept, starting with a literature review that clarifies the state-of-art of route direction generation for indoor navigation systems; then, we describe our approach analyzing step by step our solution. In Section 4, we describe our prototype for an indoor navigation system based on the Android operative system. In Section 5, we summarize the test analysis. Finally, in Section 6, we draw some conclusions and discuss future work.

2. REFERENCE DATA MODEL

Human navigation systems require storing and retrieving of different types of information. The stored information can be used for localization, path planning, generating directions, and providing location information. Depending on the approach employed by the system, this information may include floor plans, the location and description of objects in the indoor environment, locations of identifier tags or data collected using sensors [10]. Historically, routing is based on graphs since road networks can easily be described as sets of nodes and edges. One popular approach to solve this problem is to focus on the topology of rooms and build a graph to represent this topology [30]. Therefore, a very important phase in the environment representation is the simplification of the building structure, extracting the geometrical data model to support the routing algorithm. There are two different approaches used for generating a Geometric Network Model of a building (Fig. 2):

- Medial Axis Transform (MAT)-based [5];
- visibility-based [19].

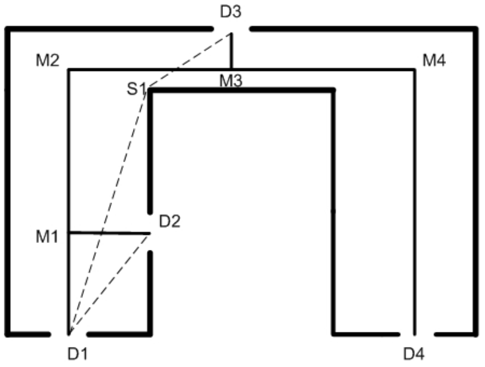


Figure 2: Comparison between MAT-based (solid line) and visibility-based (dotted line) network [17]

We choose the visibility-based approach, with a geometrical-topological network model in which *nodes* are openings (windows, doors, and passages), concave corners and furniture objects; and *transitions* are the connections between nodes which have geometrical visibility.

3. ROUTE DIRECTIONS GENERATION

The act of supporting someone by giving directions is defined as guidance; the given instructions are defined as *route directions* [25]. Route directions are primary means to guide someone in finding one's way: they are task oriented specifications of the actions to follow in order to reach the destination. The process of communicating route directions is fertile ground for scientific inquiry, being maps (especially floor plans) still the most popular presentation form. One reason is certainly their pervasive use in physical guides [4].

An important issue for maps is their orientation in relation with user's one and the surrounding space [3]. Textual instructions, instead of maps, are the most simple presentation form for navigation [24] and they are easy to create and can be used in every smartphone. In his studies in cognitive psychology, Allen analyzed route directions' production and comprehension and described all the challenges about the communication of spatial information with textual form [1] [2]. In AllenâĂŹs work, the production and comprehension of route directions is based on a *structural organization* of the route communication process, which specifically has four phases:

1. *Initiation*: ask for destination point and optionally constraints to be observed.
2. *Route description*: the respondent supplies a set of communicative statements that provides information to reach the destination. It entails two types of statement: directive and descriptive. Route descriptions involve specific components, most importantly, environmental features, delimiters, verbs of movement, and state-of-being verbs.
3. *Securing*: includes questioner's reaction to the route description, clarification queries and confirmation statements.
4. *Closure*: it is a social convention, typically verbal indications that the episode is reaching a conclusion.

3.1 Related work

Once a route (as path over the network graph) has been calculated, the first thing to think about is directional information that allows locating entities in space and defines the user's orientation. Directional relations are used in several aspects in route directions: they state the location of entities encountered along the route (like landmarks) as concerns the way finder or other entities; they announce a change of heading at decision points, e.g., represent turning actions; and they may relate these actions to an entity's location to better anchor them in space. People represent spatial knowledge, such as distances and directions, qualitatively. Since distances and directions are represented as qualitative categorical knowledge, people apply these categories also in route directions. In research on qualitative spatial reasoning, several qualitative direction models have been proposed. These models divide the two-dimensional space into (labeled) regions. These sectors map all possible angular bearings to a usually small, discrete set of categories. Against homogeneous four-sector and eight-sector direction models, Klippel et al. empirically elicited a heterogeneous direction model for turning actions in way-finding (Fig. 3) [15].

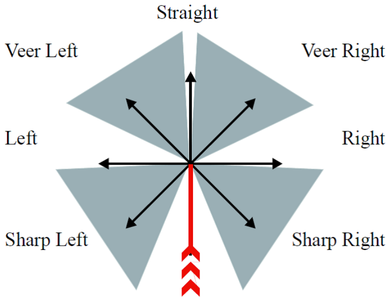


Figure 3: Qualitative direction model [15]

After the explanation of the direction concept and once planned a route, now we move to route directions communication. Since a path is divided into segments, we have a direction to follow for each segment. We have to provide instructions associated to each direction. MacMahon [21] proposed a framework in which he introduced a general structure of an instruction.

We can summarize it as follows:

- *Simple Action*: Turn, travel, verify, act, declare-goal;
- *View description*: describes visible objects with two attributes: *orientation* and *distance*.
- *Compound Action specification*: matches actions with view descriptions, using adverbs, verb objects, and adverbial clauses and prepositional phrases translated to *pre-conditions*, *while-conditions*, and *post-conditions*.
- *Qualitative reasoning properties*: the typical properties of qualitative spatial reasoning [11], *uniqueness*, *topology*, *conceptual structure*.

The usage of landmarks raises the problem of multiple instances of the same type along a leg of the route (e.g. doors, paintings, and so on). Klippel et Al. [16] proposed a process called *chunking*, summarized as selecting and merging for a route direction more than one selectable landmark into one instruction. The adopted solutions are:

- *Numerical chunking*: count items of the same class of objects and summarize them into a single instruction with a count indication.
- *Chunking based on landmarks along the route*: select landmarks close to decision points and use them to give the next instruction based on their position against the next direction (e.g. turn right before the Landmark, take the corridor opposite to the landmark, etc...). Select landmarks along route to confirm the right direction to follow.

We can also apply chunking theory to subset of edges of the route path. This technique, called *segmentation* [16] [12], consists in merging route directions of more than one path edge into a unique instruction, using *path segmentation* by splitting paths into all decision points, or using *landmark based segmentation* by splitting paths in order to create subgoals: each one is a salient landmark along the route, easy to find and see. The output of this process is often a sequence

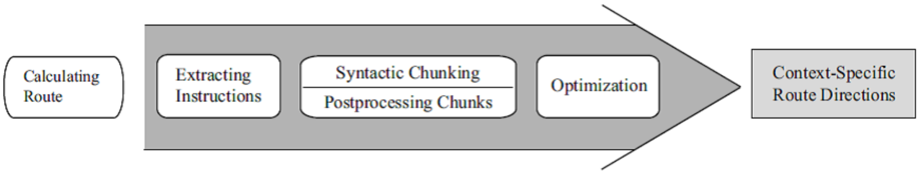


Figure 4: Overview of GUARD, the generation process for context-specific route directions [25]

of directions with a hierarchical structure, stored using a markup language to simplify the parsing of the instruction: typically, XML is the language used to describe route directions, which are still coded and not explicitly translated into natural comprehensive language, usually called *low-level route directions.*

Following the previous remarks on how to *generate low-level route direction*, Richter [25] developed *GUARD* (Generating Unambiguous, Adapted Route Directions), a computational process used for generating context-specific route directions and their interplay. It consists of four major steps depicted in Fig. 4. At first, for each decision point all applicable low-level turn instructions are generated. The individual turn instructions are then combined applying simple chunking rules; these chunks are adapted to general chunking principles in a post-processing step. Finally, in an optimization process, those chunks are chosen from the previously generated set.

The process was developed for outdoor navigation, using road network (medial axis based) datasets; since it is not easily adaptable for indoor navigation, it needs a review for pedestrian free movement and locomotion constraints. Other standard XML models for low level route instructions have been proposed by Mani et al. "SpatialML Annotation Scheme" [22] [1]: they proposed a custom data model with an XSD schema similar to the GUARD output.

The following step, after route directions generation, deals with generating natural language instructions: translating the low-level route instructions into *high level natural language* is a research field covered by the CORAL[2] project [7] [8]. Also Cuayahuitl et al. [6] improved algorithms for natural language generation starting from an XML file. Geldof's research with RPML (Route Planning Markup Language) [12] covers this field of research as well.

3.2 Proposed Algorithm

We can now explain the core problem of this research: the interaction part of an indoor Location Based Guidance Systems (iLBGS) and how we generate low level route directions that can be easily translated in various presentation forms (textual, arrows, images, etc...). We choose to follow the GUARD system approach [25], originally thought for outdoor navigation, adapted for indoor environments using a visibility based data model. Steps followed are the same as Richter's ones: extract abstract instruction for each decision node, apply chunking rules on it, optimize them using segmentation. We assume that a path between two nodes of the dataset has been computed, having at least one transition. The output is not a natural language series of instruction, but an abstract data representation, written in XML format, that encapsulates all the information needed for a complete translation into natural language.

Before starting, we introduce an example of dataset that will be used as reference for the route direction generation. Fig. 5 shows a floor of an imaginary test building, with nodes (orange dots with node number) and transitions (blue segments) and a path calculated between nodes 1 and 9 (nodes 6 & 7 are virtual doors, node 8 is a concave corner)

We apply the egocentric qualitative direction model introduced by Klippel et al. [15] (Fig. 3). As explained in previous section, we have stored in each node of the graph a direction that represents the starting user orientation while passing through the node. We put the direction model overlapping the node, centered on it, and let the node direction match with the "straight" direction. Then, we can easily determine a quantitative direction for all visible nodes matching all connecting edges with the respective belonging sector. For instance, when the user passes through node 1, (s)he has two visible nodes, node 2 that has the qualitative direction "Veer Left" and node 3 that has the qualitative direction "Veer right". Therefore, when we want to refer to node 2 (in this case, it is the next node of the path), we can use the qualitative direction model and suggest "Veer left and walk until...".

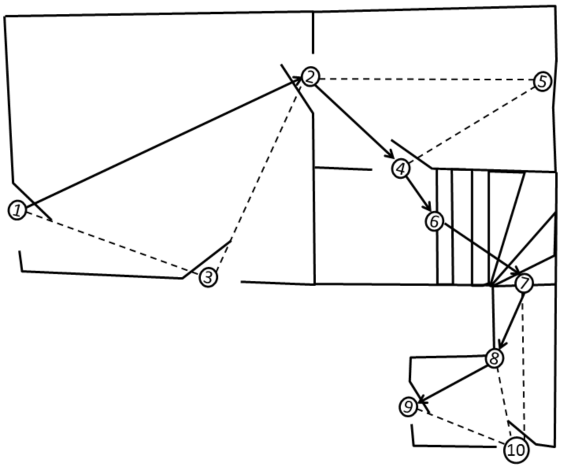


Figure 5: Example of a Building floor, with visibility graph: nodes are the circles with node number and the transitions are the dotted segments. A path between nodes 1 and 9 is highlighted with arrows

3.2.1 Route Directions Extraction

Starting from the *extraction* step and using our visibility data model, for each edge of the planned path we create

[1] http://sourceforge.net/projects/spatialml/
[2] http://web.science.mq.edu.au/ coral/

an object of the SingleEdgeDirection class. Then, we need to collect all the outgoing segments that belong to the same NavigableSpace of the path for each starting node of this list of path edges. Then, we generate an instruction to reach the next node by giving starting direction, path distance and next node name (for instance, it could be subsequently translated as "Turn right and walk for 10 meters reaching Door 1.240"), and then we add a direction for all the rest of visible nodes. We repeat this process for all the nodes in the path obtaining for each segment of the path a "next-node instruction" and a collection of visible nodes. For instance, in Fig. 5, when the user is located at node 2 and needs to reach node 4, the transition that belongs to the same space of (2,4) is only (2,5). The other outgoing transitions of node 2, which are (2,1) and (2,3), belong to a different NavigableSpace and are not considered. So, in this example, for node 2, we collect the edges (2,4) and (2,5) and define for each of them the qualitative direction, then we store the next node data about edge (2,4) separately from the others (in this case only another visible node, so only edge (2,5)) using an object of the class NextNodeInstruction, and each of the others as new object of VisibleNodeInstruction class.

3.2.2 Route Directions Chunking

With the *Chunking* step, we analyze all visible nodes directions of each node along the path, trying to highlight nodes closest to the next goal node (for instance, it could be subsequently translated as "Door 1.240 is on the Right of the Escalator") and aggregate nodes of the same Class type for numerical counting instructions (for instance, it could be subsequently translated as "Door 1.240 is after three doors on the right"). For each SingleEdgeDirection, we collect visible nodes, within a fixed radius from the next goal, in both left and right sides. We introduce ClosestChunkingInstructions that store also the belonging side of the closest node. After that, we want to group (for each SingleEdgeDirection) the visible nodes of the same class and then divide them into leftSide and rightSide referring to the next-node direction. We store each group of at least one element into another object of the class called NumericalChunkingInstruction, in which we also store the counting number of elements belonging to it.

3.2.3 Route Directions Optimization

Finally, the *Optimization* step is based on the segmentation concept. At this time, we have a direction divided into path segments, but we want to merge some of them into higher level instructions using criteria to achieve a hierarchical structure. We propose two optimization criteria:

- *room* optimization, in which all transitions in the same room (as navigable space) are merged to create a room Instruction that contains all the directions in it. This is the case of a room with a non-convex shape, without visibility between the starting and ending nodes: there are corners to be overcome to reach the next door. With this solution, the user receives detailed directions step by step, and in most cases (s)he is able to reach the next space without any other instruction. In detail, we encapsulate each SingleEdgeDirection in a new object of RoomDirection class, and group them into the same RoomDirection object when the belonging NavigableSpace of consecutive edges remains the same. We want to remark what happens between nodes 7 and 9 where edges (7,8) and (8,9) belong to the same navigable space.
- *floor* optimization, in which all VerticalUnits and HorizontalUnits directions belonging to the same collectorSpace (HorizontalSpace or VerticalSpace) are collapsed into one floor instruction as first (higher level) instruction to be performed. Usually, users with VerticalSpace movement between floors need only this instruction to reach the next HorizontalSpace (for instance a VerticalSpace instruction could be subsequently translated as "reach the third floor", and users do not need more instruction to reach the next step).

3.2.4 Route Directions Process Output

As explained before, the output of this process is an XML file that contains all these instructions and corresponds to the input of the next phase, which is the translation into natural language. In Fig. 6, we show an UML schema of the route directions and in Listing 1 a code snippet of a generated route direction.

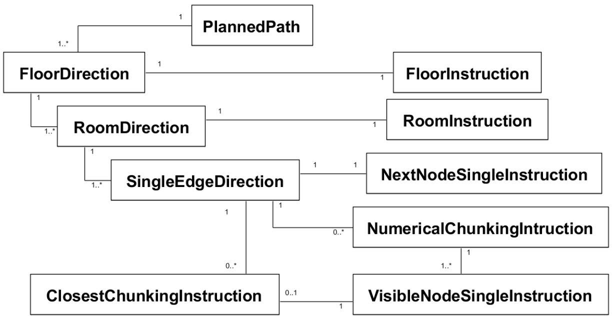
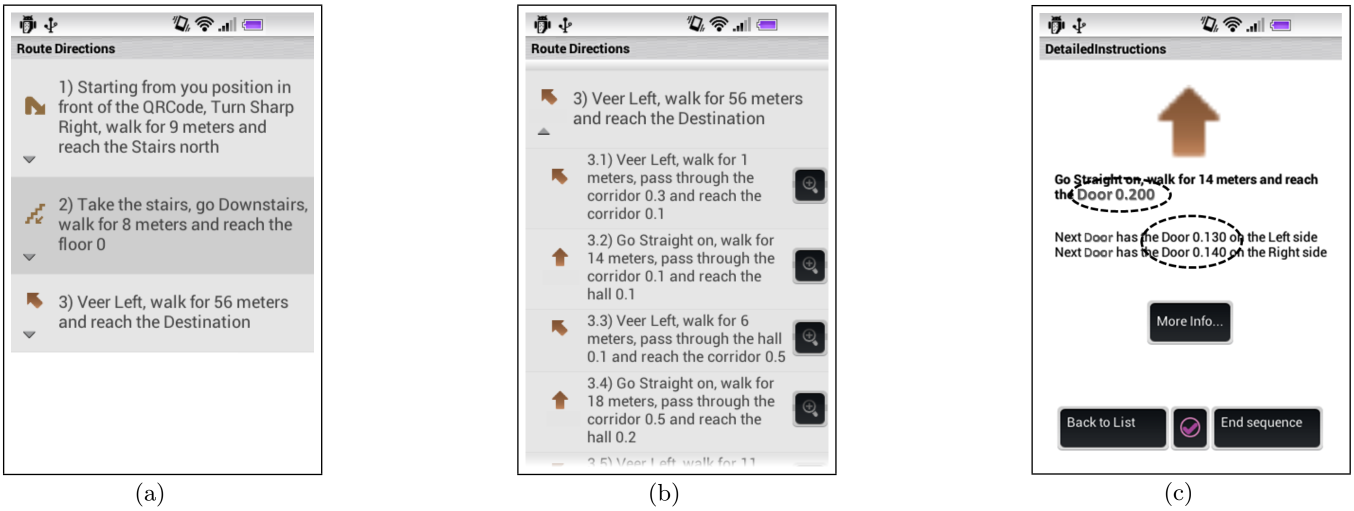


Figure 6: Route directions UML schema

```
<PlannedPath>
 <FloorDirection [...attributes hidden]>
  <FloorInstruction>
Start walking veering Right and reach the room 1.240
  </FloorInstruction>
   <RoomDirection [...attributes hidden]>
    <RoomInstruction>
Veer Right, pass through the corridor and reach the
    hall</RoomInstruction>
     <SingleEdgeDirection [...attributes hidden]>
       <NextNodeSingleInstruction>
Veer Right, walk and reach the Corridor Door
</NextNodeSingleInstruction>
        <ClosestChunkingInstruction>
Next door has the Door 1.220 on the Left side </
    ClosestChunkingInstruction>
        <NumericalChunkingIntruction>
Next door is the first visible Door from your Right
    side. </NumericalChunkingIntruction>
        <NumericalChunkingIntruction>
Next door is after 2 visible Door counting from Left
    side. </NumericalChunkingIntruction>
      </SingleEdgeDirection>
    </RoomDirection>
    [...]
 </FloorDirection>
 [...]
</PlannedPath>
```

Listing 1: Route directions XML example



Route Directions

1) Starting from you position in front of the QRCode, Turn Sharp Right, walk for 9 meters and reach the Stairs north

2) Take the stairs, go Downstairs, walk for 8 meters and reach the floor 0

3) Veer Left, walk for 56 meters and reach the Destination

(a)

Route Directions

3) Veer Left, walk for 56 meters and reach the Destination

3.1) Veer Left, walk for 1 meters, pass through the corridor 0.3 and reach the corridor 0.1

3.2) Go Straight on, walk for 14 meters, pass through the corridor 0.1 and reach the hall 0.1

3.3) Veer Left, walk for 6 meters, pass through the hall 0.1 and reach the corridor 0.5

3.4) Go Straight on, walk for 18 meters, pass through the corridor 0.5 and reach the hall 0.2

3.5) Veer Left, walk for 11

(b)

DetailedInstructions

Go Straight on, walk for 14 meters and reach the Door 0.200

Next Door has the Door 0.130 on the Left side
Next Door has the Door 0.140 on the Right side

(c)

Figure 7: IndoorNav screenshoots of the textual instruction presentation: (a) the higher level instruction, (b) the room segmentation, (c) a single edge instruction with landmarks indications highlighted: "door 1.250" as current next node and doors 0.130 and 0.140 as additional visible landmarks along the route)

4. DELEVOPED PROTOTYPE: INDOORNAV

Systems that automatically and autonomously help humans on wayfinding are called Location-Based Guidance Systems (LBGS). They are one of the most important applications of Location Based Systems (LBS) and, with the gradual maturation of ubiquitous computing and the rapid advances in mobile devices and wireless communication, they have gained increasing interest also as an important application of ubiquitous computing [14]. Following our starting requirements of a guidance system using textual instructions with the addiction of arrows and images (when needed), we have developed an Android based application called "*IndoorNav*".

In order to provide route direction service to users a web service has been developed. Based on the Apache Catalina TomCat Java server, a servlet called "RouteDirectionsServlet" answers to GET/POST web requests providing the XML file containing route directions with natural language translation.

To solve the positioning issue, we used QRCodes, assigned to each node of the graph (it translates the nodeID into a 2D image), and placed near the doors/windows/furniture objects. They are direct sensing tags with low infrastructural costs and high accuracy of user positioning. We have developed code scanning using the ZBar[3] Android library for QRCodes Image Scanning, which easily fits our needs and does not require too much time for integration in an Android application. The scanning library, when activated, automatically detects the QRCode orientation and its embedded reference code and sends the code back to the caller application.

Before explaining the interface and its basic functions, we introduce the interaction between the user, IndoorNav, and the web service. Fig. 8 illustrates the interaction flow. At first, the user scans a QRCode (1), used for understanding his starting position, then (s)he enters the ending desired point using interface interaction (and optionally disables elevators usage) (2). Then (3), clicking on the "find route" button on the interface, the IndoorNav application sends a request to the web service and (4) receives an XML response that is (5) parsed and graphically (6) rendered in the next view, so that the user can read and follow textual instructions. Whenever the user wants, (s)he can go back and scan again a QRCode near to him(her) in order to update his(her) position and update route directions.

The presentation form of route instructions, following their hierarchical structure, is shown in Fig. 7. At first level, the interface shows higher level instructions with an arrow indicating the starting direction and the distance to be covered. Clicking on one of them, the interface shows the second level list, always with arrows and distance. When we need a third level of detail, the interface provides a button on the right that shows a different interface with an arrow on the top, then the distance, the textual instruction, and optionally an image of the next target node. Scrolling down this view, the interface gives the possibility to receive other hints, such as chunking instructions and closest landmarks to the goal. All the instructions are numbered so that in the second level, users have always the reference higher level: for instance, instruction number 3) and lower level instruction 3.1).

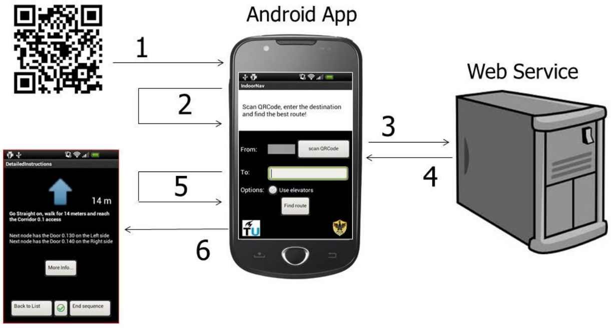
5. TEST ANALYSIS

With a handmade dataset of a testing building[4], we have performed a two-part study based on four different routes, trying to answer the question: are the route directions generated by IndoorNav understandable, allowing someone to find his (her) way? The tests were conducted involving five people and providing them an Android smartphone[5] running IndoorNav for route guidance. We have placed all QRCodes in front of the respective doors. To start each test, the user was led to the starting point and invited to follow route in-

[3] http://zbar.sourceforge.net/

[4] OTB research institute for the Built environment, Faculty of Architecture, Delft University, The Netherlands

[5] HTC Desire C



Scan QRCode, enter the destination and find the best route!

Go Straight on, walk for 14 meters and reach the Corridor 0.1 access

Next node has the Door 0.130 on the Left side
Next node has the Door 0.140 on the Right side

Figure 8: Sketch of the interaction between users and the Android application: 1) the user scans a QRCode as starting position; 2) the user enters the destination; 3) the Android app sends the route request to the web service; 4) The web service returns the route as XML file; 5) the Android app reads the received data and create the views; 6) the android app shows the output to the user

structions by scanning the starting QRcode and filling the destination field.

The users were not helped when moving and our collaborator followed from the back the users while performing the test annoting all the instructions (floors, rooms and single-edge instructions) needed to reach the destination and the number of position updates, after losing direction.

All the routes instructions started with the floor directions, so every test used all the floor instructions. At first, it is interesting to analyze how many floor directions were sufficient to guide the user to the next decision point, and how many of them required detailed room instructions to better understand the direction to follow. Secondly, we want to collect also how many room directions covered the desired additional route information and how many of them required also single edge direction to reach the next decision node. The first part of the study consists of two routes that roundly show the destination room's name, so that the users were helped in the last part of the travel by finding the room name on the doors. In the second part, we provide two routes in which the ending room names were replaced by the room ID, so that the ending rooms were unknown and users needed more accuracy to find exactly the ending room.

The analysis of the results must take into account the fact that we tested not only the generated route directions but also natural language translation and the visual prototype interface. The testing results are summarized in Table 1. All the tests had a positive outcome. Floor directions are the most useful for textual guidance (60%), particularly in vertical movements (floor changes) in which no one needed other details (100%). In spaces with non-convex shapes, room-instructions are frequently used (total percentage 25.5%). Users also lose orientation in floor directions due to unclear direction to follow, and needed to update their position scanning a near QRCode in 20% of route tests. As expected in the second part of the tests, users needed more accurate instructions due to the destination room's hidden name (single edge instruction usage has risen from 10% to 30%).

Route number	**Num. Floor instr. per route**	**Only Floor**	**Floor + Room**	**Floor + Room + SingleEdge**	**Updated position per user**
Num 1	4	12/20	7/20	1/20	0/5
Num 2	3	10/15	4/15	1/15	1/5
Summary part 1		63%	31%	6%	10%
Num 3	3	10/15	3/15	2/15	1/5
Num 4	3	7/15	3/15	5/15	2/5
Summary part 2		57%	20%	23%	30%

Table 1: Test resume of Instructions usage (all users data together) referring to the number of floor instructions per route

6. CONCLUSIONS

In this work, we were concerned with automatic generation of route directions: the proposed and implemented solution is able to produce route directions as it has been evaluated in the tests. Due to the modularity of the generation process, with low coupling between datasets, route generation and presentation form, this work may well serve as a test-bed for further empirical studies about language generation, dataset refinement or route directions comparison.

The tests have highlighted that the choice of the visibility approach solved most of the problems of data modeling, but according to Stahl et al. [29], we can improve this approach by adding a buffer (for instance 0.50m) to all the walls of the rooms. In this way, graph edges would not overlap walls/-corners, maintaining an appropriate distance from them as humans would actually do. The latter modification would better fit the case of narrow corridors, obtaining a hybrid approach closer to the MAT-based approach, and route directions generation could have more accuracy without heavy changes to our algorithm.

The objective of a positioning system with low infrastructural costs is achieved, it is a sustainable solution, but we should integrate the static positioning system (QRCodes) with other dynamic systems for a more accurate guidance while traveling. The ideas are two: firstly, we could improve IndoorNav using built-in accelerometer and gyroscope for an estimation of the position; secondly, we could introduce other direct sense positioning systems in some crucial points (previously decided by the software management) in order to catch the position while traveling and update the directions if needed.

Landmarks usage is fully integrated in the route directions, but now all the visible landmarks (as doors, windows and furniture objects) are listed in route instructions. An improvement could be a more attentive selection of landmarks to be used within a route direction.

The dynamic generation of route directions allows the system to adapt route directions to any changes of the data model in real-time: room layout changes, inaccessible doors or openings (e.g., an emergency case with fire in a corridor, with dynamic changes of the rooms accessibility), features addiction or position changing.

As future work, there are still several open issues: nodes not visible from a decision point are not taken into account in route directions generation, but during travel they could appear and disorient the user. Also path segmentation could be improved, including landmark based segmentation, so that users, while traveling in large/long rooms, could receive instructions with landmarks (visible along the path leg) as an intermediate goal to reach before arriving to the next door. The users accessibility (e.g. wheelchair) should been taken into account by the route planner, avoiding usage of graph edges (e.g. stairs) according to users requirements.

7. REFERENCES

[1] Gary L. Allen. From knowledge to words to wayfinding: Issues in the production and comprehension of route directions. In *Proceedings of the International Conference on Spatial Information Theory: A Theoretical Basis for GIS*, COSIT '97, pages 363–372, London, UK, UK, 1997. Springer-Verlag.

[2] Gary L Allen. Principles and practices for communicating route knowledge. *Applied Cognitive Psychology*, 14(4):333–359, 2000.

[3] Anthony J. Aretz and Christopher D. Wickens. The mental rotation of map displays. *Human Performance*, 5(4):303–328, 1992.

[4] Jurg Baus, Keith Cheverst, and Christian Kray. A survey of map-based mobile guides. In *Map-based Mobile Services*, pages 193–209. Springer, 2005.

[5] Harry Blum et al. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19(5):362–380, 1967.

[6] Heriberto Cuayáhuitl, Nina Dethlefs, Kai-Florian Richter, Thora Tenbrink, and John Bateman. A dialogue system for indoor wayfinding using text-based natural language. *International Journal of Computational Linguistics and Applications*, 1(1-2):285–304, 2010.

[7] Robert Dale, Sabine Geldof, and Jean-Philippe Prost. Generating more natural route descriptions. In *Proceedings of the 2002 Australasian natural language processing workshop*, pages 41–48, 2002.

[8] Robert Dale, Sabine Geldof, and Jean-Philippe Prost. Coral: Using natural language generation for navigational assistance. In *Proceedings of the 26th Australasian computer science conference-Volume 16*, pages 35–44. Australian Computer Society, Inc., 2003.

[9] Marie-Paule Daniel and Michel Denis. Spatial descriptions as navigational aids: A cognitive analysis of route directions. *Kognitionswissenschaft*, 7(1):45–52, 1998.

[10] Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. Indoor human navigation systems: A survey. *Interacting with Computers*, 25(1):21–33, 2013.

[11] Christian Freksa. Qualitative spatial reasoning. *Cognitive and linguistic aspects of geographic space*, (63):361–372, 1991.

[12] Sabine Geldof and Robert Dale. Improving route directions on mobile devices. In *ISCA Tutorial and Research Workshop (ITRW) on Multi-Modal Dialogue in Mobile Environments*, 2002.

[13] Reginald G Golledge. Human wayfinding and cognitive maps. *Wayfinding behavior: Cognitive mapping and other spatial processes*, pages 5–45, 1999.

[14] Haosheng Huang and Georg Gartner. A survey of mobile indoor navigation systems. In *Cartography in Central and Eastern Europe*, pages 305–319. Springer, 2010.

[15] Alexander Klippel, Carsten Dewey, Markus Knauff, Kai-Florian Richter, Dan R Montello, Christian Freksa, and Esther-Anna Loeliger. Direction concepts in wayfinding assistance systems. In *Workshop on Artificial Intelligence in Mobile Systems*, pages 1–8, 2004.

[16] Alexander Klippel, Stefan Hansen, Kai-Florian Richter, and Stephan Winter. Urban granularities - a data structure for cognitively ergonomic route directions. *GeoInformatica*, 13(2):223–247, 2009.

[17] L.Liu and S.Zlatanova. A "door-to-door" path-finding approach for indoor navigation. In *Proceedings of GeoInformation For Disaster Management Conference 2011*, pages 3–8, 2011.

[18] Kristin L. Lovelace, Mary Hegarty, and Daniel R. Montello. Elements of good route directions in familiar and unfamiliar environments. In Christian Freksa and David M. Mark, editors, *COSIT*, volume 1661 of *Lecture Notes in Computer Science*, pages 65–82. Springer, 1999.

[19] Tomás Lozano-Pérez and Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, October 1979.

[20] Kevin Lynch. *The image of the city*, volume 11. the MIT Press, 1960.

[21] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions In *Proc. of the 21st National Conference on Artificial Intelligence*, pages 1475-1482, 2006.

[22] Inderjeet Mani, Janet Hitzeman, and Cheryl Clark. Annotating natural language geographic references. In *proc. LREC 2008-W13 Workshop on Methodologies and Resources for Processing Spatial Language*, pages 11-15, 2008.

[23] Daniel R. Montello. Navigation. In Cambridge University, editor, *The Cambridge Handbook of Visuospatial thinking*, pages 257–294. Cambridge University Press, 2005.

[24] Verena Radoczky. How to design a pedestrian navigation system for indoor and outdoor environments. In Georg Gartner, William E. Cartwright, and Michael P. Peterson, editors, *Location Based Services and TeleCartography*, Lecture Notes in Geoinformation and Cartography, pages 301–316. Springer, 2007.

[25] Kai-Florian Richter. *Context-specific route directions: generation of cognitively motivated wayfinding instructions.* PhD thesis, University of Bremen, 2008. http://d-nb.info/987640062.

[26] Kai-Florian Richter. Prospects and challenges of landmarks in navigation services. In *Cognitive and Linguistic Aspects of Geographic Space*, pages 83–97. Springer, 2013.

[27] Tracy Ross, Andrew J. May, and Simon Thompson. The use of landmarks in pedestrian navigation instructions and the effects of context. In Stephen A. Brewster and Mark D. Dunlop, editors, *Mobile HCI*, volume 3160 of *Lecture Notes in Computer Science*, pages 300–304. Springer, 2004.

[28] Alexander W Siegel and Sheldon H White. The development of spatial representations of large-scale environments. *Advances in child development and behavior*, 10:9–55, 1975.

[29] Christoph Stahl. New perspectives on built environment models for pedestrian navigation. *Spatial Cognition 2008 Poster Proceedings*, pages 016–08, 2008.

[30] Horst Steuer. High precision 3D indoor routing on reduced visibility graphs. In *Progress in Location-Based Services*, pages 265–275. Springer, 2013.

[31] Edgar-Philipp Stoffel. *Hierarchical graphs as organisational principle and spatial model applied to pedestrian indoor navigation.* PhD thesis, Ludwig Maximilians University Munich, 2009. http://d-nb.info/100055080X.