Context-Aware Indoor Navigation

Fernando Lyardet[1] and Diego Wong Szeto[2] and Erwin Aitenbichler[2]

[1] SAP Research CEC Darmstadt, Darmstadt 64283, Germany
[2] Darmstadt University of Technology, Hochschultrasse 10, 64289 Darmstadt, Germany

Abstract. Over the past few years, several technological advances have been made to enable locating people in indoor settings, where way finding is something we do on a daily basis. In a similar way as it happened with GPS and today's popular outdoor navigation systems, indoor navigation is set to become one of the first, truly ubiquitous services that will make our living and working environments intelligent. Two critical characteristics of human way finding are destination choice and path selection. This work focuses on the latter, which traditionally has been assumed to be the result of minimizing procedures such as selecting the shortest path, the quickest or the least costly path. However, this path approximations are not necessarily the most natural paths. Taking advantage of context-aware information sources, this paper presents an easy to deploy context-aware indoor navigation system, together with an efficient spatial representation, and novel approach for path adaptation to help people find their destination according to their preferences and contextual information. We tested our system in one building with several users to estimate first an assessment of preference values, and later to compare how the paths suggested by our system correspond to those people would actually follow. The positive results of this evaluation confirm the suitability of our models and algorithms.

1 Introduction

Many people, especially business travelers, are often confronted with the situation that they are visiting new places and have to figure out how to navigate to the places where they want to go. Because path finding is a frequently occurring real-world problem, many studies and applications have been done on this subject over the last years. These studies provide several interesting insights, e.g., primary criteria for human path selection [1], or how people often prefer routes that are easier to describe and follow over routes that are optimal in a theoretical sense. An important indicator of perceived easier-to-follow routes are those with fewer instructions. In order to achieve this goal, different non-optimal algorithms that provide simpler paths were proposed, e.g., by Liu [2] and Richter [3]. However, these approaches were focused on street navigation, where the movement of people and vehicles follow the rules of street orientation.

We believe an aspect that is hardly covered by current research literature is the need of adapting to the current situation of the user, and to the condition of the environment in indoor navigation applications. The variation of conditions people may encounter in environments such as airports, museums, corporate campuses, shopping malls, factories, etc. is so diverse that they play a deciding role on the criteria to which path would be the one people would actually follow.

For this reason, we have developed a user-centric approach for indoor navigation. In this paper, we first describe our hybrid indoor location model required to efficiently calculate possible paths between two given locations. Then we present a novel user model that includes information about physical capabilities, access rights, and a flexible modeling of user preferences based on Multiattribute Decision Theory (MAUT). We also describe in detail, how these preferences are analyzed and how context information is used to select the most appropriate path for the user.

Finally, we describe the architecture of the implemented CoINS system which can work with several different local positioning systems (LPS), such as WLAN, Ubisense, or Elpas. A particularly interesting deployment scenario is based on camera smartphones and QR-Codes for locating users. This solution provides a low-entry technology threshold and allows to use CoINS on any recent smartphone without the need to deploy any proprietary software components on the phone.

1.1 CoINS Overview

The aim of the Context-aware Indoor Navigation System (CoINS) is to provide efficient user navigation in buildings with a strong emphasis on the “human factor”. When considering the human as part of the system, the term *efficient* does not longer simply correspond to the shortest path calculated by some mathematical method. To efficiently navigate users to their destinations, it is also vital that they can quickly comprehend and execute the navigation instructions they receive from the navigation system. For example, a good route description would consist of a low number of turns, turns would be at ”landmarks” the user can easily identify, and would always clearly indicate the directions in which the user is supposed to walk. The user model and the path selection algorithm presented in this paper allow to include such considerations into route planning.

An important subcomponent of CoINS is *path calculation*, which only considers the ”geometric” aspects of route planning. In our earlier work [4] we only used models and algorithms based on quadtrees [5]. This limited the scalability of the system to larger buildings. The present CoINS system is able to efficiently find the geometrically shortest paths because of the following two extensions: First, the symbolic part of our hybrid location model is based on a hierarchical graph. Second, we improved the search path algorithm with an heuristic to further reduce the search space. These two factors allow to keep the single graphs small to reduce pathfinding complexity to $O(|L|-1)+O(|V_i|)$ in the average case, and $O(|V_i|^2)$ in the worst scenario (see 2.1).

2 World Model

The CoINS world model is a hybrid model that combines symbolic graph-based models with geometric models. A symbolic model is required for indoor navigation, because *room numbers*, *corridor names*, *floor numbers*, etc. have an intuitive semantic to users. Using geometric coordinates for end-user interaction would not be suitable. The geometric model is needed for determining the shortest paths and to obtain orientation information for guiding users into the correct directions.

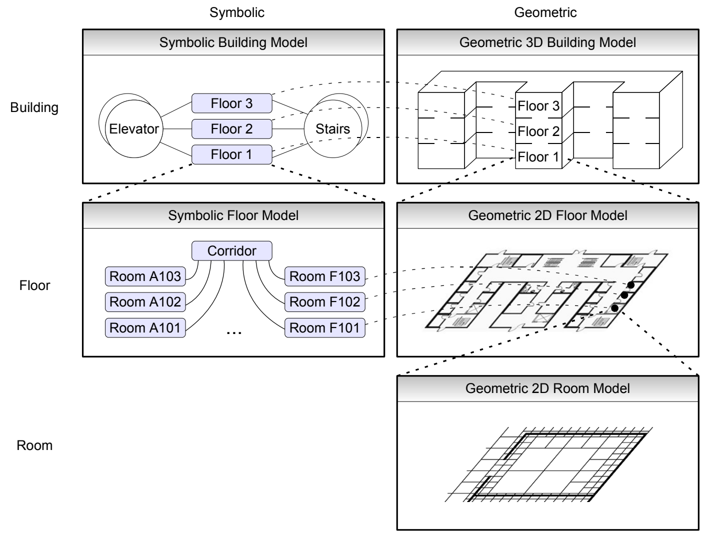


Fig. 1. Hybrid world model of CoINS.

The world model serves two main purposes. First, it supports transformations between geometric coordinates and symbolic locations and vice versa. When a 3D tracking system is used that provides geometric coordinates to locate users, the model must be able to transform this coordinate into a symbolic location, such as a room number. The pathfinding algorithm of CoINS starts with the symbolic models to create a coarse plan of the route. After that, the geometric models are used for fine-planning. At the transition between using symbolic and geometric models, the world model must also support transforming symbolic into geometric coordinates.

Second, the model enables efficient pathfinding. The design of the CoINS world model has been refined over several iterations to ensure that the search sets are as small as possible and that the basic relations needed by the pathfinding algorithm can be checked efficiently. In most cases, users will mostly move in two dimensions. Movements in the height dimension usually only occur when changing floors, which is modeled by using separate maps for each floor.

The world model of CoINS is shown in Figure 1. According to modeling level and model type, the world model consists of five parts:

Geometric Building Model: This is a detailed geometric 3D model of the entire building. All other model parts are created at design time from this model. At run time, this model is not needed by the pathfinding algorithm of CoINS. However, if a 3D local positioning system is used in the building, the model is needed to transform metric coordinates to symbolic locations.

Geometric Floor Model: The floor model is a geometric 2D model of a single floor. Here, the term *floor* denotes some part of the building at the same level that is connected. It does not necessarily have to correspond to the whole story of a building. On the same floor, users will only be able to move in two dimensions. When guiding a

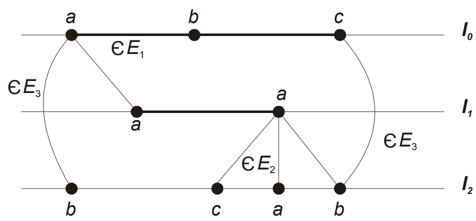


Fig. 2. A vertex-labeled universal hierarchical graph.

user on a floor, a 2D representation is sufficient. The reduction to 2D space simplifies path calculation and increases performance significantly.

Geometric Room Model: The room model is a geometric 3D model of a room or a corridor. While it is not necessary to model rooms in detail, the geometry of corridors is important for finding shortest paths and to obtain direction information for navigation.

Symbolic Building Model: This is a symbolic graph-based model of the whole building. Its nodes are *floors* and *connecting structures*, such as elevators and stairs. Each floor node holds a reference to its symbolic floor model.

Symbolic Floor Model: This is a symbolic graph-based model of a floor. Its nodes are *rooms* and *corridors*, connected by entrances and exits.

2.1 Graph-theoretical Model

This new model can be formalized using a recently specified graph class called universal hierarchical graphs [6]. In [6], undirected universal hierarchical graphs have been introduced to calculate the topological entropy of such graphs. After this, the graph class has been extended to the directed case [7]. Also, it was shown that these graphs can be efficiently classified by transforming them into certain property strings capturing important structural information.

In the following, we extend the definition given in [6] to vertex-labelled graphs and state a special edge type specification.

Definition 1. *Let*

$$V := \{v_{0,1}, v_{0,2}, \ldots, v_{0,|V_0|}, v_{1,1}, v_{1,2}, \ldots, v_{1,|V_1|}, \\ v_{2,1}, v_{2,2}, \ldots, v_{2,|V_2|}, \ldots, v_{|L|-1,1}, v_{|L|-1,2}, \ldots, v_{|L|-1,|V_{|L|-1}}\}, \quad (1)$$

be the vertex set, $|V| < \infty$ *and let*

$$A_V^U := \{l_1, l_2, \ldots, l_{|A_V^U|}\}, \quad (2)$$

be the unique vertex alphabet. $l_V : V \longrightarrow A_V^U$ *represent the corresponding vertex labeling function.* $L := \{l_0, l_1, \ldots, l_{|L|-1}\}$ *defines the level set.* $|L|$ *denotes its cardinality.* $\mathcal{L} : V \longrightarrow L$ *is a surjective mapping that is called a multi level function if it assigns to each vertex an element of the level set* L*. The graph* $U = (V, E)$ *is called a vertex-labelled universal hierarchical graph* $\Leftrightarrow$ *its edge set can be represented by the union* $E := E_1 \cup E_2 \cup E_3$*. We specify the sets* E_i *as follows:*

– E_1 *denotes the set of horizontal Across-edges. A horizontal Across-edge does not change a level* i.
– E_2 *denotes the set of edges which change exactly one level. .*
– E_3 *denotes the set of edges which overjump at least one level.*

The set of undirected labelled universal hierarchical graphs is denoted by $\mathcal{G}_{LUHG}$.

As an example, Figure (4) shows an undirected labelled universal hierarchical graph. Here, it holds $A_V^U := \{a, b, c\}$. For example, it is $l_V(v_{0,1}) = a$ and $l_V(v_{2,4}) = b$. We see that our objects we deal with can be described by graphs $U \in \mathcal{G}_{LUHG}$. The relationship to general graphs can be understood by recalling that an arbitrary graph is not necessarily hierarchical (the hierarchies must be induced). Generally, hierarchies can be induced by selecting a distinct root and applying algorithms based on shortest path, see, e.g., [8]. In our case, the hierarchy is given naturally because the building we want to model is hierarchical.

In the following, we consider a special class $\mathcal{G}^{\star}_{LUHG}$ of undirected labelled universal hierarchical graphs such that E_2 can be written as

$$E_2 = \{\{v_{0,i_0}, v_{1,i_1}\}, \{v_{1,i_1}, v_{2,i_2}\}, \dots, \{v_{|L|-2,i_{|L|-2}}, v_{|L|-1,i_{|L|-1}}\}\} \cup E_2^{\star}, \qquad (3)$$

where $1 \leq i_0 \leq |V_0|, 1 \leq i_1 \leq |V_1|, \dots, 1 \leq i_{|L|-1} \leq |V_{|L|} - 1|$. Speaking informally, that means we assume that there always exists at least one directed path starting from at a vertex $v_{0,i_0} \in V$ and ending at $v_{|L|-1,i_{|L|-1}} \in V$ (e.g. as it is the case with stairs connecting floors). $E_2^{\star}$ is assumed to contain the remaining edges $e \in E_2$. By using this assumption, we easily derive the following assertion.

Theorem 1. *Let* $U = (V, E) \in \mathcal{G}^{\star}_{LUHG}$. *The time complexity for finding an arbitrary vertex* $v \in V$ *is* $O(|L| - 1) + O(|V_i|)$.

Proof: Let $U = (V, E) \in \mathcal{G}^{\star}_{LUHG}$ and arbitrary $v \in V$. Because there exists a path $P = (v_{0,i_0}, v_{1,i_1}, \dots, v_{|L|-1,i_{|L|-1}})$, the worst case time complexity to calculate the distance for vertices on this path is consequently $O(|L| - 1)$. The time complexity for finding a vertex on level i is $O(|V_i|)$.

Corollary 1. *Let* $U = (V, E) \in \mathcal{G}^{\star}_{LUHG}$. *In case there exists a vertex on each level* i *which connects all remaining vertices on this level, the time complexity for finding an arbitrary vertex* $v \in V$ *is* $O(|L| - 1) + O(1)$.

Theorem 2. *Let* $U = (V, E) \in \mathcal{G}^{\star}_{LUHG}$. *Without the assumptions we made in Theorem (1), the time complexity for finding an arbitrary vertex* $v \in V$ *(starting from another vertex) is* $O(|V|^2)$.

Proof: Let $U = (V, E) \in \mathcal{G}^{\star}_{LUHG}$. We always assume that our graphs we deal with are connected. To start from an arbitrary vertex and calculating the shortest distance to another vertex, any shortest path algorithm can be used, e.g., Dijkstra-algorithm [8]. For this, the complete adjacency matrix must be parsed. But this requires time complexity $O(|V|^2)$.

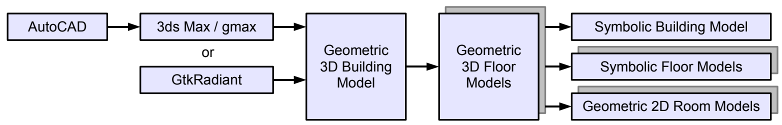


Fig. 3. Generation of the World Model.

2.2 Generation of the World Model

Figure 3 shows the generation of the CoINS world model and its parts.

CoINS currently uses the Quake III map format [9] as input for the world descriptions, since it is widely known and open source tools are available to create world descriptions. To create models for CoINS we have used the GtkRadiant kit [10] and Autodesk Gmax [11] which allows us to import AutoCad drawings. A detailed discussion of describing the world using Quake maps, and the further processing required to map into 2D maps and identify rooms is presented in [4].

3 A Model for User Centric Adaptation

The user model we have developed and applied in our indoor navigation technology combines three key aspects besides user identification data: physical capabilities, user preferences, and location access rights.

3.1 Physical Capabilities

Physical capabilities are represented using the International Classification of Functioning, Disability and Health (ICF) of the World Health Organization (WHO) [12]. The ICF has been developed by the WHO as a framework for measuring health and disability, and proposes several types of qualifiers depending on the kind of impairment evaluation. For body functions, a qualifier shows the presence of an impairment which is measured on a five point scale: no impairment, mild, moderate, severe and complete. For the sake of simplicity, only voluntary leg movement-related functions involved in walking are considered, although this model can be extended with other body functions.

3.2 Location Access Rights

Location access rights control the entrance of users to certain locations or places. In CoINS, a user can enter a location, if her access rights are available. This is modeled according to the Role Based Access Control (RBAC) model [13]. A user profile is a set of roles. Each role has a set of permissions that grant or restrict access to locations. User profiles and roles are managed within a session, in which user, profiles, roles and the permissions are assigned. Concretely,

USER:	{users}	SESSION:	$\subseteq USER \times PROFILE \times 2^{ROLE}$
PROFILE:	{Profiles}	PERMISSION:	{ granted, denied }
ROLE:	{Roles}	ROLE:	USER: $\times PROFILE \mapsto 2^{ROLE}$
LOCATION:	{Locations}	PERMIT:	$ROLE \times LOCATION \mapsto PERMISSION$

3.3 Preferences

The user preferences influence the selection of paths. These preferences describe the desire or predisposition of a user in favor of something. In this case, the preferences take the attributes of paths into account. Some examples of such attributes are temperature, luminosity, distance and crowdedness (Table 1). Since such preferences can be arbitrary, we need to measured them with a numerical scale so that the preference is quantifiable [14]. The range of the preference scale varies from 0 to 100, where 0 is equivalent to least and 100 the most preferable. In Table 2, we show different user pref-

(a)

Distance (m)	**Assessment Value**
0	100
10	95
30	85
60	45
125	20
200	3

(b)

Number of turns	**Assessment Value**
0	100
5	80
10	40
18	1

(c)

ADRT range	**Category**	**Assessment Value**
0°C - 10°C	Low	100
11°C - 20°C	Medium	40
20°C - higher	High	0

Table 1. User preference assessment for (a) distance, (b) turns, and (c) temperature variation.

erences in their natural scales together with their upper and lower bounds. The upper bound values of Path length and number of turns is different for every building, therefore an estimation was done by calculating the longest path possible in the building minimizing repetition of places. On the other hand, other maximum values such as path crowd density and variance are standards already documented elsewhere [15].

On certain occasions the construction of a natural scale is not possible. For instance, the beauty of a path is hard to measure with a scale, since it involves aspects such as color combination, layout, among others. For those scenarios, it is necessary to substitute the attribute with other(s) that are easier to quantify. For instance, beauty can be replaced with attributes like homogeneity of the locations layouts and the height of walls.

3.4 Modeling User Preferences with Multiattribute Utility Theory

This model presents a Multiattribute Utility Theory (MAUT)-based architecture that enables decision making according to user interests (Table 3). Each user has a different setting of preferences, and each with a different priority to the user. The strength of a preference is given by an assessment function f. Let's define F as the set of all assessment functions f. Nevertheless, every user has different preferences when faced

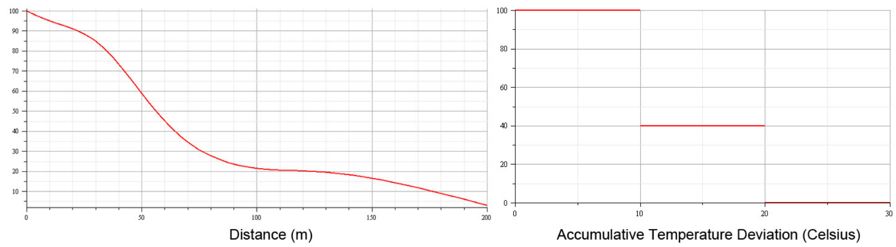


Fig. 4. The first curve shows an example of a polynomial approximation with distance values in meters. Preferences can also be categorized as ”low”, ”medium” and ”high”, resulting in the step function on the right.

Attribute	**Lower bound**	**Upper bound**
Path length (m)	0	3780
Number of turns	0	246
Number of visual aids	0	75
Acum. temperature deviation (°C)	0	100
Path crowd density $(persons/m^2)$	0	6
Crowd density variance $(persons/m^2)^2$	0	0.2

Table 2. Natural scales of path preference attributes.

in several different situations. As a result, an assessment function f holds if certain conditions apply. Let's define E as our set of possible events e. An event e represents a situation of the user like for instance, the user is carrying heavy luggage or the location is on fire.

Depending on the application domain, every assessment function f quantifies the preference strength of some criterion c. These criteria comprise several aspects concerning the application. In the domain of indoor navigation, criteria such as *path length*, *number of turns* or *crowdedness* are relevant.

Each of these criteria may also be subject to user specific trade-off behavior. One way to solve this is to assign priority weights to the assessment functions, which is conventionally done in standard Multiattribute Utility Theory (MAUT) techniques.

F	= {f | f is an assessment function}	COND	= {STATE, ACTIVITY, LOCATION, TIME}
E	= {e | e is an event}	STATE	= { happy, busy, available, . . . }
X	= {x | x can be assessed with some f ∈ F }	ACTIVITY	= { run, read, work, . . . }
where	$f : \mathcal{P}(E) \times X \rightarrow \{0, \ldots, 100\}$	LOCATION	= {room a036, kitchen, . . . }
And:			
E	= (SUBJ, COND)	TIME	= {starttime, endtime}
SUBJ	= {user 1, user 2, room a036, . . . }		

Table 3. Multiattribute Utility Theory definitions, and attribute examples.

Even though preference strength is measured by assessment functions, the preference priority can change when events occur. Someone walking in an airport with empty hands could probably care less about the path length than after picking his luggage. In that situation, the importance weight of the path length criteria increases from not being empty handed and having luggage. Therefore, preference priority weights should be adjustable to different circumstances when events happen. An Event Listener notifies the client about an incoming event e_i (1). Upon receiving the event notification, the client updates the preference functions by forwarding the notification message to every criteria. As a result, every criteria knows what events affects them and by how much the preference priority is increased or decreased.

4 Path Calculation and Selection

To select the most suitable path for a specific user we use the Simple Multi-Attribute Rating Technique (SMART). Under this technique, every path can be described by individual preference attributes and through the value functions of each single attribute, the preference strength can be measured.

The process of decision making can be influenced by the user's consideration of what attributes are more important or relevant to him. For this reason, importance weights are given to attributes in order to model this kind of situation. The importance weight alters the output of an attribute's value function.

The measured preference strength of every single attribute of a path can be aggregated to return an overall path evaluation value. This resulting value will be later used to compare against other possible paths.

4.1 Assessment of Preferences with Simple Multiattribute Rating Technique (SMART)

Our optimal path selection problem using SMART is formalized as follows:

Let P be the set of all feasible paths, then $P = \{p|p$ is a path over the location model$\}$ and a_n all attributes with importance weights w_n and value functions v_n. Then, the optimal path $p \in P$ maximizes

$eval(p) = aggregate(w_i(v_i(a_i(p))))$ for i = 1,. . . , number of attributes

The following five steps describe how the path selection using the SMART method is conducted:

1. **Define the feasible paths and attributes for evaluation.** All feasible paths are generated. Given that locations are connected as a graph in the location model, all paths can be generated by traversing the graph using a depth first search, A* or Dijkstra. User constraints (e.g. physical disabilities) can be used at this stage to avoid traversing unnecessary paths thus filtering them out from the selection process.
2. **Assign the importance weights to the attributes.** The importance of attributes to the user are established by assigning importance weights to attributes. These weights are normalized,which means in this case that the sum of all weights equals

Attribute	Importance	Path 1	Path 2	Path 3
Path length	0.4	39.31 (65m)	51.67 (55m)	18.6 (138m)
Number of visual aids	0.2	99.29 (19)	96.52 (18)	3.09 (50)
ADRT	0.1	100 (4°C)	100 (6C)	0 (29°C)
Number of turns	0.1	94.03 (2)	94.03 (2)	80 (5)
Path Crowd Density	0.05	89.7 (0.11p/m^2)	91.66 (0.09p/m^2)	87.69 (0.13p/m^2)
Crowd Density Variance	0.15	99.22 (0.012$(p/m^2)^2$)	86.51 (0.008$(p/m^2)^2$)	98.86 (0.017$(p/m^2)^2$)
Evaluation value		74.35	**76.93**	35.27

Table 4. Assessment of path attributes. Path 2 is the selected path.

one. This weight assignation allows an ordering of attributes according to their relevance and therefore the preference strength of an attribute with a high importance weight will have a higher impact on the path evaluation than those with lower. Importance weights can be constants or functions. The latter is necessary when no constant ordering of path attributes can be modeled. For example, if the importance depends on the preference strength of some other path attributes.

3. **For every path, assess the path attributes separately.** At this step the iteration of value measurement over all attributes for every path generated in the first step is performed.
4. **Calculate evaluation value by aggregating all attribute assessments under consideration of importance weights.** In the fourth step, assessment values of paths attributes from step three are aggregated. Different aggregation models exist to approach this. For SMART, a very common model is the additive, which sums up all assessment values from attributes multiplied by its respective importance weight, i.e. $eval(p) = \sum_{i=1}^{n} w_i d(a_i(p))$.
5. **Select the path with the best evaluation value.**In the last step, path selection can be done by searching for the path with a target evaluation value. How the target evaluation value is defined can depend on how the value scales were created during path attribute assessment. For example, if the assessment functions map into numerical values between 0 and 100 where 0 means the least desirable and 100 the most, then path selection will concentrate on paths with highest evaluation values.

5 CoINS Architecture

Figure 5 shows the architecture of the CoINS system. The *Presentation* component of CoINS can be either accessed as web interface through a web server or from rich clients. The web-based solution has the advantage that no software deployment is necessary on clients, but location tracking is limited to QR codes or purely infrastructure-based solutions. In contrast, rich clients can provide more customized user interfaces and support additional local positioning systems.

The *Path Calculation* component determines all possible routes from a given source location to a given destination location. In addition, it provides the geometric length of each route. This component only considers the location model.

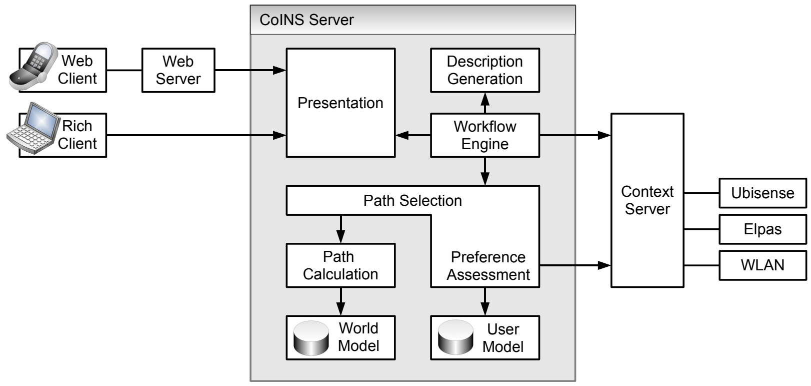


Fig. 5. CoINS System Architecture.

The *Path Selection* component uses *Preference Assessment* to calculate all non-location related metrics for all candidate routes and then selects the best route. The candidate routes have been previously determined by Path Calculation. Preference Assessment uses the route, the user model, and the current context as inputs.

The whole process is controlled by a small *Workflow Engine*. Using a workflow allows us to decouple the guidance process from the CoINS core system. CoINS server, the Context Server, the location tracking systems, and rich clients are based on our communication middleware MundoCore. MundoCore and the Context Server are described in more detail in the following sections.

5.1 MundoCore Middleware and Software Architecture

MundoCore [16] is an open-source communication middleware specifically designed for AmI applications. It supports automatic discovery of peers in the network and provides a peer-to-peer publish/subscribe system. The middleware has mechanisms to handle network failures and handovers between different networks, allowing users to transparently continue accessing services in such conditions. In addition, MundoCore implements an Object Request Broker on top of publish/subscribe to achieve location and execution transparency.

Therefore, programmers of an application do not need to know about the address of its communication partners (a server, for instance). The distribution of client and server components can be determined at deployment time or even changed at runtime.

For the implementation of CoINS, we adopted a service-oriented architecture, because it allows deploying application components according to different scenarios and improve their reuse by other applications. For example, if a client only supports a web browser, all components of CoINS can run on a server in the Internet. A more powerful client such as an Ultra Mobile PC could already run the whole CoINS system locally.

5.2 The Mundo Context Server

The Mundo Context Server [17] is responsible for transforming raw sensor readings into information that is meaningful to applications. It is an application-independent and reusable component that aims to decouple applications from sensors such that sensors become interchangeable and that new sensors can be added without requiring to change applications. The context server provides the following functionality:

- Interpreting data received from sensors and transforming this data into a common representation.
- Maintaining a geometric world model of the environment and supporting geometric operations and queries.
- Inferring "higher-level" context from "lower-level" context.
- Notifying applications when certain context properties change.
- Storing histories of sensed and inferred context and supporting queries in those histories.

CoINS uses the Context Server to track the locations of users. Because the Context Server provides an abstraction layer above the physical sensors, CoINS can use standardized queries and does not have to be aware of the underlying sensor technology. The Context Server is based on the notion of widgets to process context information. A processing chain for a location sensor usually consist of the following steps:

1. **Sensor:** The first widget in the chain obtains the raw sensor data. In most cases, it is a MundoCore Subscriber that receives messages from a remote sensor in the network.
2. **Normalization:** Normalization transforms a system-specific message into a normalized, system-neutral message. The classes of normalized messages are composed by "multiple inheritance" and implement an ID part and a location part.
3. **Transformation:** This step handles entity/ID transformations, e.g. to translate from tags to users, or from base stations to rooms, or positions between coordinate systems.
4. **Inference:** This step infers relations from messages. A relation typically consists of two entities and describes a relationship between those two entities. Relations are expressed as RDF statements.
5. **Storage:** The relation store stores relations and is based on the Tuplespace concept.
6. **Query/Subscription:** Applications can query the relation store or subscribe to the store. A query operation returns all relations matching a specified pattern. When an application subscribes to the store, it will be notified each time a tuple matching the specified pattern is inserted into the space.

The current implementation of the CoINS system can work with the following tracking systems:

- **Ubisense:** Ubisense [18] is an UWB-based high-resolution 3D tracking system. For this system, the processing chain consists of a normalization to a common representation for 3D tracking systems, coordinate system transformation, tag-to-user ID mapping, transformation to symbolic coordinates using the *Geometric Building Model*.

– **Elpas:** Elpas [19] is a badge-based system that combines IR, LF, and RF. It can cover entire buildings and depending on the amount of infrastructure deployed, its accuracy varies between 25m and 2m. Elpas provides symbolic location in a system-specific format. The processing chain consists of a normalization to a common representation for badge-based systems, tag-to-user ID mapping, and sensor-to-room ID mapping.
– **WLAN:** WLAN-based tracking [20] can utilize WLAN infrastructures that often already exist. WLAN-based systems can reach an accuracy of about 2-3m, given that the necessary RSS fingerprint maps are accurate.
– **QR:** Quick Response (QR)-Codes are two-dimensional barcodes that were standardized by ISO in the year 2000 (ISO/IEC 18004). To date, QR reading software is available for almost any smartphone. QR codes can be used to create *physical world hyperlinks*: A user having a camera phone can scan the image of a QR Code causing the phone's browser to launch and redirect to the programmed URL. When using QR codes as a location source, the physical location of the QR code is encoded into the URL. The context server is then used to map location codes to symbolic locations.

The Context Server would support even more tracking systems. However, these systems either cover areas too small for indoor navigation applications (IRIS [21]) or they are not accurate enough (Cellphone tracking [22]).

6 User Guidance

6.1 Customizable Guidance Process

The CoINS system decouples the process of guiding a user by including a small, embeddable workflow engine called Micro workflow [23]. In this way, both the supporting services and the process of indoor user guidance can be further modified. Another advantage is that this approach also enforced further separations of responsibilities in order to allow a possible change of the workflow engine if required. Figure 6 shows the implementation of a guidance workflow.

6.2 Giving Directions to The User

Different techniques and requirements have been studied for guiding users indoors with mobile platforms [24, 25]. Guiding techniques includes floor maps, schemas, and spatial landmarks [26] together with their spatial relationships to help build a "mental trip" on the guided user.

The user guidance in the current version of CoINS is realized through a visual modality, where instructions are displayed in the user's terminal. We have so far developed 2 different versions for the user terminal. The latest one is a "thin" client, in order to support the common situation where no software deployment can be done on the user terminal.

In a typical scenario, a user enters the building and points with the mobile's camera to any of the QR-Codes available in every door (Figure 7). A URL is encoded that

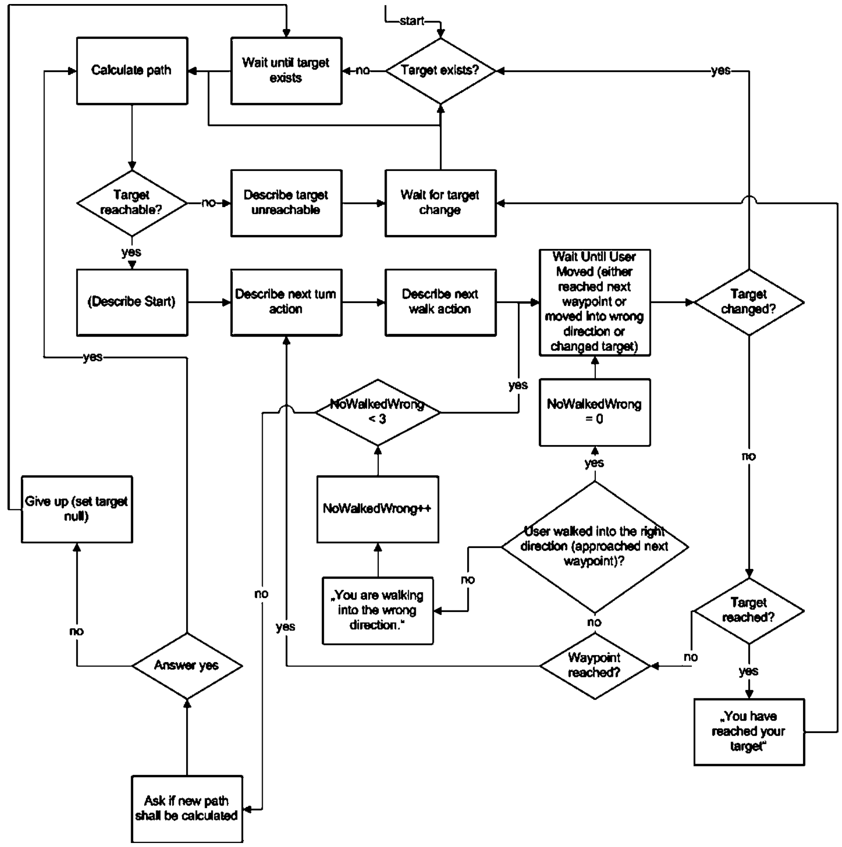


Fig. 6. Workflow Implementation Overview.

takes the user to a welcome page where information of the current location (people to contact, office hours, contact information) is displayed, and the question is the user requires directions. When the user selects this option, she can either type the name of the person or room number, and the system starts the guiding process (current location is assumed to be where the QR-Code was read).

The user continues receiving instructions everytime she requires it. Along the way, the user may feel insecure about the current location and how to proceed. In such cases, just by scanning any QR-Code, CoINS resumes que navigation from the current possition to the originally specified.

Indoor location systems are seldom installed in large scale settings, and can be either very expensive or impractical following the building construction characteristics. Therefore, the choice of a standard such as QR-codes appears as a natural solution. Clearly, the limitations of the thin client and QR-Codes as a location systems makes it difficult for proactive guidance. However, we are currently experimenting with new approaches to overcome these restrictions.

Fig. 7. A user scanning a QR-Code.

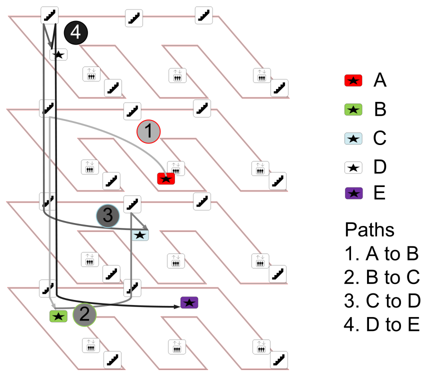
7 Evaluation

Two different experiments have been carried out to study the paths people follow to reach a given destination in indoor settings. The purpose of the first evaluation was to gather information to find a common profile for the experiment, that is, a set of assessment functions (three of them are shown in table 1(a),(b) and (c)) and a ranking of the different criteria to estimate their weight in the path selection assessment procedure (shown in table 4).

For the first experiment 8 subjects where scheduled, all of them familiar with the Piloty Building at the University of Darmstadt, a 4-story building, that offers in every floor five different stairs and 2 elevators to change floors (see figure 5). A set of pairs of origins and destinations were carefully selected, and the subjects were asked to carry out simple tasks that implied walking between the selected destinations(e.g. *"You need to pick up a form in office XXX"*, *"Take the form to Mr. YYY in office XYZ"*, *"Go to the library"*,*"Get a Coffee at the Bistro"*, etc.). To start the experiment, subjects were met at the origin for their assigned route and then were read the appropriate directions. Afterwards, the subjects began walking to the given destinations as a part of the assignments given. The routes every participant followed were recorded in separate maps of the area, and all possible paths were coded to record every time a particular path was followed. Subjects then completed a questionnaire on the criteria they used in selecting a path.

The second experiment involved 9 subjects (5 female, 4 male) that did not participate in the first experiment. We chose the same set of destinations as in the first experiment, and assigned the same tasks to the subjects, following the first experiment's procedure. We also used the data and user preference profile information gathered during the first experiment for the algorithm to calculate for every participant the suggested path, in order to take into account the contextual characteristics at the moment where the experiment would take place, and compare it to the one actually followed by the subject. The results of this first evaluation of the system are shown in table 6. These results suggests that the proposed paths by system matches those chosen by people at a particular contextual setting. In table 5 we show an schema of the building, with the proposed paths by the system and the alternatives performed by the users. Although encouraging, this study must also be performed in our future work in large indoor scenarios such as airports, where users may have more options of movement according to the context.

(a)



(b)

Subj.	M/F	A to B Path Chosen	B to C Path Chosen	C to D Path Chosen	D to E Path Chosen
1	M	Y	Y	Y	Y
2	M	Y	Y	Y	Y
3	M	Y	Y	N	Y
4	F	N	Y	Y	Y
5	F	Y	Y	Y	Y
6	F	Y	Y	N	Y
7	F	Y	Y	Y	Y
8	M	N	Y	N	Y
9	F	Y	Y	Y	Y

Table 5. (a) Shows the structure of the Piloty building, and the paths proposed for evaluation. (b)Table shows for each participant and path, whether the path followed by the subject matches the path proposed by the system.

	Path 1	Path 2	Path 3	Path 4	Σ
♯ Dijkstra's generated Paths	24	9	24	57	
♯ CoINS heuristic's generated Paths	9	9	1	1	
Users that followed path as predicted	7	9	6	9	**31**
Users that followed a different path	2	0	3	0	**5**

Table 6. Generated paths and evaluation results.

In Table 6 we show the result of the reduced search space done by the heuristic implemented in CoINS allowed a reduction of potential paths to be analyzed of more than 82% (114 found by the general Dikjstra vs. 20 from CoINS spearch space reduction + Dijkstra). The results of the user behavior are encouraging: more that 86,1% of the users actually followed the paths suggested by the algorithm.

8 Related Work

Human way finding has been studied from many different perspectives. Understanding human way finding and navigation has been done in architecture and geography studies. How human convey information to guide and orient others with verbal instructions [27]. Different studies have shown the relative importance of path optimality: people would accept a less optimal (in terms of distance) path, in favor of routes that are potentially easier to describe or follow. Perhaps the closest related system is by Tsetsos et al. [28] presenting Ontonav , a system that introduces the use of ontologies in indoor navigation to describe and reason over the user preferences, and annotate the world model, in order to determine the path users must follow. Finally, Yao et al. [29] also proposes a navigation system using geo-coded QR-codes for individuals with cognitive impairments.

Other relevant systems are Drishti developed by Lisa Ran, Sumi Helal and Steve Moore in [30]. Drishti is an integrated indoor/outdoor navigation system for visually impaired people that uses commercial off the shelf soft- and hardware. And finally the Navio Project [31] from the Vienna University of Technology, a pedestrian navigation system in mixed indoor and outdoor environment. It addresses several important aspects like positioning, route planning, and communication of pedestrian navigation services.

9 Summary and Future Work

We have developed CoINS, a context-aware indoor navigation system with efficient route planning. A key element to the efficient calculation of routes between two given locations is the hybrid location model presented in this paper. We have also discussed the application of QR codes for easy to deploy indoor navigation, without having to deploy any specific software on the end-user terminal. Finally, the selection of the most suitable path is done through a process of assessing user preferences according to the context information.

As future work we plan to add automatic customization of the preferences to each user, as well as the set of preferences to be evaluated and the ranking among them. On the technology side, we will extend the current web-based user client to a rich application, and take advantage of compass-enabled smartphones.

References

1. Golledge, R.G.: Path selection and route preference in human navigation: A progress report. In: COSIT. (1995) 207–222
2. Liu, B.: Intelligent Route Finding: Combining Knowledge, Cases and an Efficient Search Algorithm. In: Proceedings of ECAI96, 12th European Conference on Artificial Intelligence, John Wiley & Sons, Ltd (1996) 149–155.
3. Richter, K.F., Duckham, M.: Simplest Instructions: Finding Easy-to-Describe Routes for Navigation. In: Lecture Notes in Computer Science (GIScience 2008). (2008)
4. Lyardet, F., Grimmer, J., Mühlhäuser, M.: CoINS: Context sensitive Indoor Navigation System. In: Proceedings of IEEE International Symposium on Multimedia (ISM), Los Alamitos, CA, USA, IEEE Press (2006) 30–37
5. Samet, H.: The Quadtree and Related Hierarchical Structures. ACM Computing Surveys (CSUR) (June 1984)
6. Emmert-Streib, F., Dehmer, M.: Information theoretic measures of uhg graphs with low computational complexity. Applied Mathematics and Computation **9** (2007) 1783–1794
7. Dehmer, M., Emmert-Streib, F.: Structural similarity of directed universal hierarchical graphs: A low computational complexity approach. Applied Mathematics and Computation **194** (2007) 7–20
8. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Mathematik (51) (1950) 161–166

9. Proudfoot, K.: Unofficial Quake 3 Map Specs. http://graphics.stanford.edu/ kekoa/q3/ (March 2000)
10. GTK: Radiant. http://www.qeradiant.com (March 2006) last visited: 03.07.2008.
11. Autodesk: Gmax. http://www.autodesk.com/gmax (2008) last visited: 03.07.2008.
12. World Health Organization: International Classification of Functioning, Disability and Health. http://www.who.int/classifications/icf/en last visited: 03.07.2008.
13. Ferraiolo, D., Kuhn, R.: Role-Based Access Control. In: Proceedings of the 15th National Computer Security Conference. (1992) 554–563
14. Edwards, W., von Winterfeldt, D.: Decision Analysis And Behavioral Research. Cambridge University Press (1986)
15. Neufert, E., Neufert, P.: Architect's Data. Blackwell Science (2000)
16. Aitenbichler, E., Kangasharju, J., Mühlhäuser, M.: MundoCore: A Light-weight Infrastructure for Pervasive Computing. Pervasive and Mobile Computing **3**(4) (August 2007) 332–361 doi:10.1016/j.pmcj.2007.04.002.
17. Aitenbichler, E., Lyardet, F., Mühlhäuser, M.: Designing and Implementing Smart Spaces. Cepis Upgrade **VIII**(4) (August 2007) 31–37
18. Ubisense: Homepage. The Smart Space Company. http://www.ubisense.net/ (last visited: 03.07.2008) (2008)
19. Visonic Technologies: Elpas. http://www.visonictech.com/elpas.html (2007) last visited: 03.07.2008.
20. Song, Y.: In-House Location Tracking. Master's thesis, Darmstadt University of Technology (2002)
21. Aitenbichler, E., Mühlhäuser, M.: An IR Local Positioning System for Smart Items and Devices. In: Proceedings of the 23^{rd} IEEE International Conference on Distributed Computing Systems Workshops (IWSAWC03), IEEE Computer Society (May 2003) 334–339
22. Hartl, A.: A Provider-Independent, Proactive Service for Location Sensing in Cellular Networks. In: GTGKVS Fachgespräch (Online Proceedings). (2005)
23. Manolescu, D.: Workflow Enactment with Continuation and Future Objects. In: OOPSLA'02, ACM (November 2002)
24. Butz, A., Baus, J., Krüger, A., Lohse, M.: A Hybrid Indoor Navigation System. In: IUI2001: International Conference on Intelligent User Interfaces, New York, ACM Press (January 2001) 25–32
25. Fraunhofer: Messe Navigator. http://www.iis.fraunhofer.de/ec/navigation/indoor/projekte/messe (accessed at April 2006)
26. Sorrows, M., Hirtle, S.: The Nature of Landmarks for Real and Electronic Spaces. In Freksa, C., Mark, D.M., eds.: COSIT '99. Volume 1661., Springer (1999) 37–50
27. Streeter, L.A., Vitello, D., Wonsiewicz, S.A.: How to Tell People Where to Go: Comparing Navigational Aids. International Journal of Man-Machine Studies **22**(5) (1985) 549–562
28. Tsetsos, V., Anagnostopoulos, C., Kikiras, P., Hasiotis, P., Hadjiefthymiades, S.: A Human-Centered Semantic Navigation System for Indoor Environments. Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on (July 2005) 146–155
29. Chang, Y.J., Tsai, S.K., Chang, Y.S., Wang, T.Y.: A Novel Wayfinding System Based on Geo-Coded QR-Codes for Individuals With Cognitive Impairments. In: Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility, New York, NY, USA, ACM (2007) 231–232
30. Ran, L., Helal, S., Moore, S.: Drishti: An integrated indoor/outdoor blind navigation system and service. In: PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04), Washington, DC, USA, IEEE Computer Society (2004) 23
31. Gartner, G., Frank, A., Retscher, G.: Pedestrian navigation system in mixed indoor outdoor environment the navio project (2004)