Smart Signs: Showing the way in Smart Surroundings

Maria Lijding[1], Hartmut Benz[2], Nirvana Meratnia[1], and Paul Havinga[1]

University of Twente[(1)], Twente Institute for Wireless and Mobile Communications[(2)]

Abstract. This paper presents a *context-aware guidance and messaging system* for large buildings and surrounding venues. Smart Signs are a new type of electronic door- and way-sign based on wireless sensor networks. Smart Signs present *in-situ* personalized guidance and messages, are ubiquitous, and easy to understand. They combine the easiness of use of traditional static signs with the flexibility and reactiveness of navigation systems. The Smart Signs system uses context information such as user's mobility limitations, the weather, and possible emergency situations to improve guidance and messaging.
Minimal infrastructure requirements and a simple deployment tool make it feasible to easily deploy a Smart Signs system on demand.
An important design issue of the Smart Signs system is privacy: the system secures communication links, does not track users, allow almost complete anonymous use, and prevent the system to be used as a tool for spying on users.

1 Introduction

Consider the workshop of a big research project or a commercial seminar. After the last morning session, the participants head to the restaurant for lunch, which will be served in another building. When the participants leave the meeting rooms they consult the Smart Signs attached next to the doors and in corridors. As a participant approaches a Smart Sign, it displays a group-message (*Follow the arrows to the restaurant*) and an arrow pointing at the right direction. Some participants like Peter, who is currently on clutches, and Maya, who uses a wheelchair, receive a personal message and guiding arrow, which routes them via paths without stairs.

Halfway to the restaurant, a sudden fierce rain shower triggers the Smart Signs system to re-route the participants through a longer route to protect them from getting wet.

One of the participants, Toni, suffers from epilepsy. He is wearing a context-aware monitoring device that connects him to a health-care center that warns him of an imminent attack and sends help when necessary. During an evening session, Toni faints and his monitoring device contacts the health-care center and announces the emergency locally. This 112-like call for help can be picked up by devices of doctors and paramedics in the vicinity—similar to someone shouting *Is there a doctor in the room?* The Smart Signs system reacts to this emergency by flashing a call for help and show guidance to get to Toni. Another emergency procedure of the Smart Signs system is to guide everybody to the nearest fire exit and mustering station in situations when the building must be evacuated.

To realize this scenario, Maria, the organizer, earlier walked around the premises and attached Smart Signs at important doors, corridors, and outside locations. The Smart Signs are battery-driven and automatically form a multi-hop network without further configuration. While deploying the signs, Maria entered some additional meta-data for each of the nodes on her PDA (location on a floorplan, logical address, name). All participant have received Smart Tags on arrival, which identifies them to the Smart Signs. Figure 1 illustrates examples of our current Smart Signs and Smart Tags.

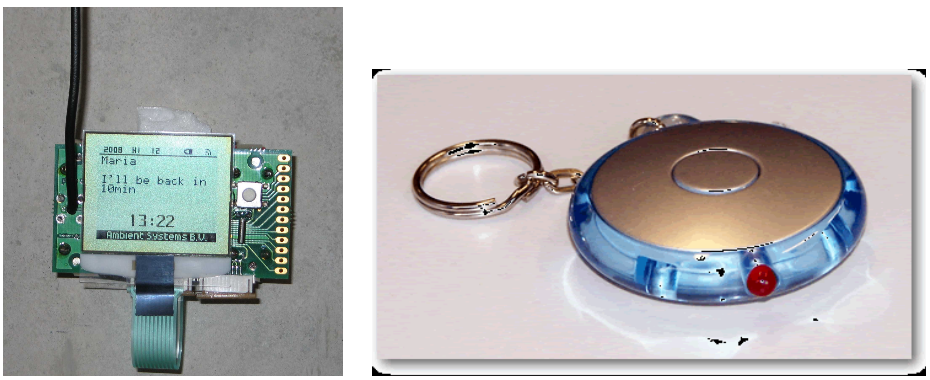

Fig. 1. Our current Smart Sign (left) and Smart Tag encased in a keyring (right)

The Smart Signs are new types of electronic door- and way-signs based on wireless sensor networks (WSN). A WSN consists of large numbers of cooperating small-scale nodes capable of limited computation, wireless multi-hop communication, sensing, and actuating. Our wireless sensor nodes have small displays attached. The Smart Signs can be deployed as part of a heterogeneous network, with more powerful devices and bigger displays (e.g., laptops or PCs) in crucial points such as the reception of a building or registration point of a conference. Other type of interfaces such as audio or haptic output can also be supported.

Smart Signs provide guidance and messaging functionality. They combine the advantages of signage, which subsumes traditional directional signs and messages, with the reactiveness and flexibility of personal services, such as navigational support, and SMS. The Smart Signs system also uses context information such as user's mobility limitations, the weather, and possible emergency situations like fire or medical needs to optimize routes and messaging.

Signage is easy to understand—mainly because of our familiarity with it—presents information *in situ*, is ubiquitous and easy to ignore when not relevant. Another advantage of signage that is generally overlooked is that it is anonymous, because neither the signs, nor the people who install them know who uses the signs. Thus, traditional signs provide absolute privacy from the point of view of the user.

However, traditional signs have important shortcomings. Using signage for navigation requires that a user knows milestones on the route to his destination, because the signs cannot provide directions to every destination. Another shortcoming of signage is that the signs cannot adapt automatically to real time changes—e.g. a lift temporarily out of order—and it is difficult to keep them up-to-date. Last, but not least, static signs lack the capacity to show information only to the interested parties, and to be displayed only at the right moment. For example a Post-it saying ‘Hand in your work to the secretary’ at a researcher’s door is clearly not meant for everyone passing by, but just for his students.

Navigation systems use the destination of the users to guide them through the best path taking into account context information, for instance traffic jams. Using a good navigation system, a driver should no longer fear getting lost, because the system can guide him from any location to his destination, even if he decides to take a detour for sightseeing or if he gets distracted and misses some directions.

A shortcoming of Smart Signs compared to traditional signs is, that the users have to sacrifice some privacy by telling the system where they want to go or to whom they want to post messages. We have tried to limit the privacy violation as much as possible by making privacy an important design issue in Smart Signs. We secured the communication links to prevent eavesdropping and designed a system that is privacy aware: we do not track users, allow almost complete anonymous use, and prevent the system to be used as a tool for spying on users.

An overview of recent related work is provided in Section 2. An extensive description of the main architectural components and deployment of the Smart Signs system follows in Section 3. We describe a number of alternative user interfaces in Section 4. Section 5 details our privacy-aware design, and Section 6 gives some concluding remarks and future work.

2 Related Work

Since the beginning of Ubiquitous Computing research much work has been done to offer people information about their surroundings through personal devices. Cyberguide [1] is an early application to show tourists their location on a map and give information about objects in their vicinity. GUIDE [5] supports tourists using audio, maps, texts, and pictures. Specific systems have also been designed to support visits to museums and exhibitions. Ciavarella et al. [10] present a museum visitor assistant using PDAs, which shows the room floorplan and an audio explains the highlights. These system generally lack navigational support.

2.1 Indoor Navigation Systems

Indoor navigation systems are on an early development stage. Baus [4] describes IRREAL, an indoor navigation system for handheld devices. IRREAL uses infrared beacons in significant places of the building to inform users about their current position, orientation, and some nearby objects. The deployment is costly and difficult because each beacon requires its own PC and infrared requires line of sight to communicate.

Miu designed and implemented an indoor mobile navigation system called CricketNav [21] using the Cricket [12] indoor location system developed at the MIT. CricketNav uses RF and ultrasound to determine the location of the user. It then uses a map of the venue to find the shortest path to the user's destination and show it on the user's PDA. Despite the success of the system to achieve reasonably high positional accuracy and preserving the user's privacy, the CricketNav requires large amount of extra hardware to be installed by the venue owners.

The newly developed navigation system ARIADNE [3] combines various technologies such as GPS, WLAN, and RFID to provide indoor and outdoors positioning services. Navigational guidance is given to the users on their laptops, which requires an additional RFID tag. In addition, ARIADNE imposes extra burdens on users by forcing them to scan their tag with RFID readers on the venue. Without manual scanning, the system cannot provide guidance.

2.2 User Interfaces

Navigational systems running on handheld devices that require users to look at the device 'head down' while walking are not easy to use. Chincholle [9] shows that the limited screen space of mobile devices make maps and text instructions difficult to use on the move.

As an alternative, some researchers propose to use haptic output to provide guidance. GentleGuide, a prototype developed by Bosman et al. [6], uses vibration on the wrists to inform people about directions to follow. The authors concluded that a very simple coding scheme indicating only left/right turns, stop and wrong directions works best. Zelek et al. [28] use two small video cameras wired to the user's personal device to capture environmental objects and provide feedback through a special glove worn by the user. Mechanical vibrators in the glove navigate the user through the building. Recently, Van Erp et al. [13] proposed using a vibrotactile waist belt.

An interface prototype of a 'rotating compass' is proposed by Rukzio et al. [26], in which a display is installed at every decision point to constantly show different directions. When the direction shown on the display matches the direction that a nearby user should follow, the mobile phone of the user vibrates. The mobile phone needs to know the route that the user has to follow and must be able to synchronize with all the displays in the venue.

There is much research on non-visual interfaces to support wayfinding for visually impaired people. Ross et al. [25] have evaluated different types of audio output in outdoor scenarios with a focus on street crossing. In the Chatty Environment [11], objects in a building are tagged with passive or active tags (e.g. RFID, Wireless Sensor Nodes). The user's device detects the tags and announces the identity of the object using audio.

2.3 Context-aware Messaging

Systems such as [18, 2] let users create contents (or annotations) that are linked to locations. Other users in the vicinity of the location can discover the annotation on their handheld device. Using GPS technology, ComMotion [20] and Stick-e [7] provide location-based messaging functionality through personal devices. The difference between the two lies in the ability of the former to provide personal messages, while the latter provides public messages. Sohn et al. [27] design a system called Place-Its for posting location-based reminders on mobile phones. Based on their findings, they concluded that location-based messages are often a reminders to perform an activity. Iachello et al. [14] share this point of view.

Cheverst et al. [8] propose to use bluetooth in mobile phones to interact with displays situated at the venue. In this case, people are able to leave messages and even photos at office doors to be shown to specific persons. The same concept is used by Nichols et al. [23] to use a desktop program to post the availability status of employees on the display at their office door.

3 System Architecture

Figure 2 gives an overview of the main components of the Smart Signs system, related services, and client applications. Most components are realized as *Services*, which can be accessed via Java APIs, Jini, or as a Web Service. We implement the core functionality of the components in the form of *Business Rules* to achieve flexibility for the interaction with context and context reasoners. Most of our business rules are currently hard-coded in objects, or as table-based rules in the Smart Signs.

The main component of the Smart Signs system is the *Smart Signs Service*, which is primarily a container aggregating three sub-services. It provides the API for guidance and message requests. The Smart Signs Service runs on a central server. Clients running on kiosk-terminals, PDAs, Smart Phones etc. can access this service via a Web interface or Java RMI.[1] Not shown are the functions for user and group management and to access map-related data about the premises (e.g., floorplans, names of locations).

The *World Model* is a database containing topological and architectural data about the venue, including access rights, and names of locations. See Section 3.1 for more details.

[1] We do not discuss client applications for posting guidance and message requests in this paper.

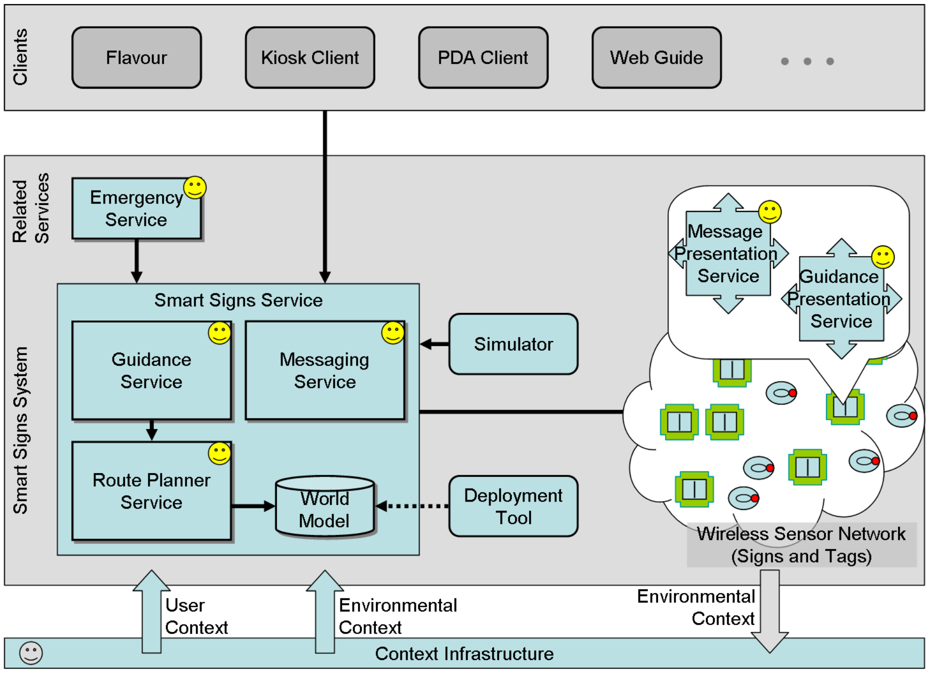

Fig. 2. High-level architecture and deployment of the Smart Signs system. Significant Business Rules are indicated by a smily icon

The *Route Planner Service* computes routes in the building and premises using the *World Model*. The route planner can consider several parameters (i.e., user preferences, environmental conditions) to optimize the routes for specific circumstances. The service produces compressed routing tables that are used by the *Guidance Presentation Service* on the Smart Signs to show directions. Business rules in the route planner determine which and how parameters are translated into cost of a path. Section 3.2 discusses this service in more detail.

The *Guidance Service* accepts and manages guidance requests during their lifetime and distributes them to the respective *Guidance Presentation Service* in the Smart Signs. Business rules in this service determine which request to accept, how to obtain the user's preferences regarding routing from his/her context, the lifetime of requests, etc. Guidance uses the Route Planner Service to compute the routing tables for the Smart Signs. It refreshes these tables automatically when the context of a user changes or the Route Planner indicates changes in the environmental context it uses. Section 3.3 discusses this service in more detail.

The *Messaging Service* accepts and manages message requests during their lifetime and distributes them to the respective *Message Presentation Service* in the Smart Signs. Messages can be posted to all or to individual Smart Signs. They can be shown to everyone, a group, or only a single user. Message presentation can also be context sensitive (e.g., during working hours, when it is raining).

The Smart Signs are small wireless sensor nodes that can run multiple services. To act as Smart Signs, the nodes run two services: the *Guidance Presentation Service* and the *Message Presentation Service*. They can also be used by other applications at the same time. For example a node can have temperature, humidity and light sensors and provide context information to the context infrastructure. This context information can also be used locally to decide when a message has to be shown. Section 3.4 discusses the Guidance Presentation Service in further detail.

The *Deployment Tool* is an application to create the World Model for the Smart Signs system. The application offers a simple drag-and-drop user interface to assign real-world locations to Smart Signs based on floorplans of the venue (Fig. 3 on page 9). Key to the success of the Smart Signs system will be easy deployment. Physical deployment of smart signs is easy because they can be attached to walls, doors or other suitable surfaces. They do not require a power connection (at present running on batteries for up to 15 weeks) and can be placed up to 150 meters apart (under optimal conditions). Our heterogenous network enables us to bridge longer distances between nodes.

The *Smart Signs system Simulator* allows to simulate the functions of the deployed Smart Signs network and the context received via the context infrastructure. The simulator client allows the operator to post guidance and message request to the Smart Signs Service and to control the context available to the service. It visualizes the Smart Signs network on a floorplan of the building(s) and the operator can drag user-figures around the floorplan and immediately see which guidance and messages these users would get (see Fig. 4 on page 12).

The *Emergency Service* is a related service implementing the emergency behavior outlined in the introduction. In its current implementation it listens to 112-like emergency calls on the premises of the Smart Signs system on a special emergency context channel. When a personal emergency is detected, the Emergency Processor posts a general guidance request to the location of the emergency and posts a general messaging request asking for a doctor to help. When an environmental emergency is detected, the Emergency Processor cancels all existing guidance requests, posts a guidance request for everyone to the nearest fire exit, and a general message describing the emergency.

The *Context Infrastructure* provides the generic interface to access context and to be informed about changes in the user's preferences and environmental conditions. Business rules in the context infrastructure reason about context. Context reasoning provides flexible mechanisms to derive user preferences relevant to the Smart Signs system. The context infrastructure may even provide a generic rule engine for context reasoning. Use of business rules in the context infrastructure is a research topic by itself and will not be discussed in this paper.

3.1 World Model

To provide guidance and messaging functionality in a scalable, efficient, and flexible manner, we assign a logical address to every relevant location and area in the venue, as well as objects of interest (e.g., printers, coffee machines). Naturally, all these also have a name and a location associated. The structure of our world model is inspired by IP [24].

We divide the venue in *Spaces* (comparable to subnets in IP) that contain *Addressable Locations* (similar to an IP host). A space can also contain other spaces (e.g., a floor in a building). Some addressable locations have *Smart Signs* attached, which can guide users (similar to routers in IP) and show messages. If a Smart Sign belongs to intersecting spaces, we call it an *Access Point* [2].

Access Points can have associated *access rights* to indicate who can transit in each direction and under which circumstances. This functionality is especially useful when representing doors. Defining access rights in each direction provides rich semantics such as expressing that outside working hours everyone can leave a building, but only authorized personnel can enter.

Smart Signs have associated *posting rights* that indicate who is allowed to post on a sign. For example, the people in a meeting room may get posting rights on the Smart Sign on the room's door during the meeting, and only a short time before and after the meeting. Posting rights can be defined at a space level and are inherited by all Smart Signs in the space.

Access and posting rights are defined for owners (e.g. the people who work in a room), groups (different groups may have different rights) and others. The rights take a value from the set of situations {Always, During working hours, When use granted, During emergency, Never}.

We use a hierarchical addressing mechanism, where each address is a 32-bit number $Bb.Ff.D.Rr.O$ (4 bits per letter). Bb indicates a building. Ff represents a major area in a building, as for example a floor or a staircase. D represents a minor division such as an area assigned to a department. Rr indicates a room number, and O represents an object such as a printer, a Smart Sign, the cash at the canteen, etc.

Spaces are defined by an address, a mask, and a 3D-space. As in IP, masks can be applied to any address to determine whether they are part of a space. A space can also have associated properties and posting rights, which are inherited by the locations and spaces included in it. Edge properties are further discussed in Section 3.2.

The world model of a venue can be easily defined using the *Deployment Tool*. The operator only needs to specify the logical structure of the premises using *spaces*. These spaces are then used by the tool to automatically suggest a hierarchical address space. The tool provides a drag-and-drop interface to manage elements representing Spaces, Addressable Locations, Smart Signs, and Access Points. Figure 3 shows part of a floor-plan as seen in the deployment tool. On the floor-plan there are addressable locations marked by an arrow pointing down,

[2] Access Points provide all the functionality of Smart Signs

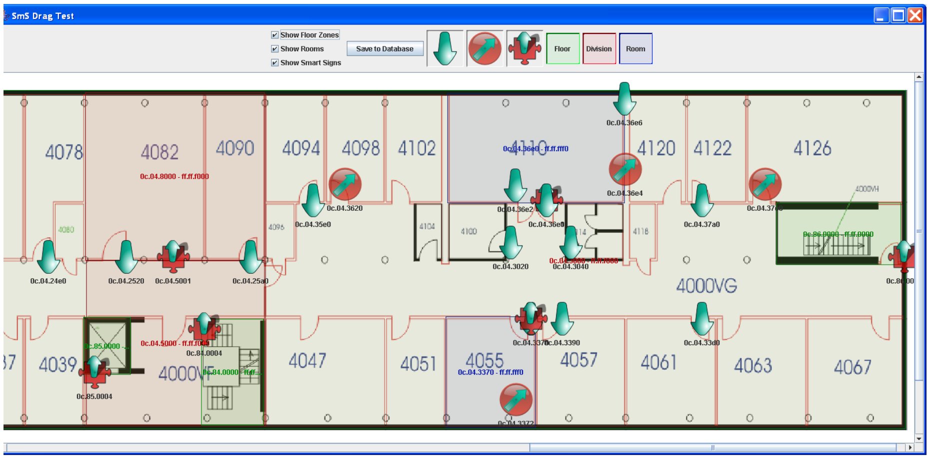

Fig. 3. Example of the World Model and support provided by the deployment tool

Smart Signs marked by a circle with an arrow, and Access Points marked by a puzzle piece with an up-arrow. In the area of the stairs and lifts, it can be clearly seen how spaces intersect. Rooms 4110 and 4055 have objects (printer, coffee machine, etc.) that can be addressed and some even have Smart Signs attached. There are Access Points for these two rooms with access rights assigned.

3.2 Route Planner Service

The *Route Planner* computes routes for every Smart Sign to all possible destinations. The goal of the route planner is to produce compact routing tables that can be sent to the Smart Signs. The Smart Signs can then decide autonomously how to guide people. Using the addressing and masking mechanism described in the previous section routing tables can be compressed very efficiently.

The route planner builds a *directed weighted graph* from the world model. The world model does not contain edges, which makes defining and maintaining the model easier. We deduce edges automatically in such a way that the resulting graph is connected, while at the same time the number of edges is kept small. In each space, the Smart Signs are connected with each other bidirectionally. In turn they have unidirectional edges to the addressable locations in the space. If there is no Smart Sign to cover the addressable locations at one space, they will be covered at a higher level.

Edges have properties, which are inherited from the space they belong to. Outgoing edges of an Access Point also inherit its appropriate access rights.

When computing the weights of the graph, the preferences and access rights of the current user are taken into account, which in turn are used to interpret the environmental context. Weight computation is triggered by the *Guidance*

Service when a user makes a request, when the preferences or context of a user with an active request changes, or when the environmental context changes.

The basic weight of an edge is the geographical distance between the nodes plus the additional cost derived from the *context*. The additional cost may be infinite, which makes an edge not eligible in any path.

Here we discuss the set of edge properties, user preferences, and environmental context we are currently handling.

Edge Properties

- Placement of edge: {Indoors, Outdoors, Outdoors Protected from rain}
- Inclination: {Flat, Stairs, Lift, Ramp}
- Means of transport that the edge can manage: Set of {Walking, Wheel-chair, Clutches, Bicycle, Car, Motorbike}.
- Risk factor: Set of (Situation x Risk *value*), where the situation is described with the following values {Always, During working hours, Outside working hours, At night, When wet, When fire in building, When fire in the area} and the risk value is in the range [0,1].
- Accessibility: only valid for outgoing edges of an Access Point.

User parameters . The user parameters are given by the user's preferences and access rights. The preferences may be quasi-static, provided by the user directly, or derived from his context using user-specific rules. We consider at present the following parameters:

- Affinity to stairs: The value is in the range [0,1], where 0 indicates stairs should be definitely avoided.
- Affinity to lifts: The value is in the range [0,1], where 0 indicates lifts should be definitely avoided.
- Transportation availability: Indicates the transportation that the user has at his disposition. This parameter is given by a set of values from {Walking, Wheel-chair, Clutches, Bicycle, Car, Motorbike}. The values of this parameter override the affinity to stairs or lifts (e.g. if the user is on a wheel-chair cannot use stairs, it does not matter how much he loves stairs).
- Weather preferences: Indicates what is acceptable and ideal weather for the user to be outdoors. The preferences are a combination of temperature, precipitation, brightness, and humidity.
- Access Rights: Indicates the access rights that the user has to all the different Access Points in the venue. This access rights are also context dependent.

Context Information

- Time: Time is a trigger to changes in context. For example some doors may only be accessible to everyone during working hours, and only to authorized personnel at other times. Depending on time also the risk factor of certain edges may change.

- Weather: Weather information is important in order to determine the weight of using outdoor edges. An outdoor edge becomes less attractive (and, thus, heavier) when it is raining hard or it is very cold. When determining the cost of an edge, this information is used in combination with the user preferences. The weather information is composed of the following parameters: temperature, precipitation, brightness, and humidity.
- Transient situation: This is information about situations which affect some part of the venue. For example, a lift out of order, current or expected traffic, wet or just waxed floor, a blocked door, a closed canteen, etc.
- Emergency situations: These situations are also transient, but their relevance is higher. An example is a medical emergency that requires medical personnel to be guided to the person needing help, even if they are not carrying tags.

3.2.1 Compressed Routing Tables

For each Smart Sign the route planner computes *routing* and *adjacency tables*. An entry in the routing table contains: [Destination (Address), Next Step (Address), Cost], while an entry in the adjacency table is: [Neighbor (Address), Direction, Cost]. Direction indicates the direction the user needs to follow towards his destination at that point and is given by a bearing, distance on XY-coordinates, and a height difference (if the user needs to go up or down).

Without compression the routing tables grow linearly with the number of addressable locations. It is important to compress the routing tables as much as possible, because Smart Signs do not have much memory. The μNodes we are currently using, have 48 KB of flash memory, 10 KB of RAM, and 4 Mbit of external flash/EPROM. Moreover, the tables have to be communicated to the nodes, and communication is the most energy-consuming part of a WSN.

The Smart Signs only need to know the direction towards a destination. Therefore, an entry for a *compressed routing table* is simply: [Destination (Adrress-Mask), Direction]. The compressed routing tables are obtained from combining the routing and adjacency tables, and creating entries for spaces that have a same next step in the routing table. Luckily, most destinations in a space require to follow the same direction, thus, we can very efficiently compress the routing tables using masking.

To be able to scale even more and cover thousands of destinations, in the future we are going to use hierarchical routing as it is done in IP. We will define *administrative domains* (similar to Autonomous Systems in IP) for example at building level. We can then use an internal routing protocol within a domain and an external routing protocol between domains. This will also allow multiple operators to provide the world model for their domain and to define domain-specific context sources.

3.3 Guidance Service

The task of the *Guidance Service* is to receive guidance requests from users and groups, and to send the guidance rules for that request to the *Guidance*

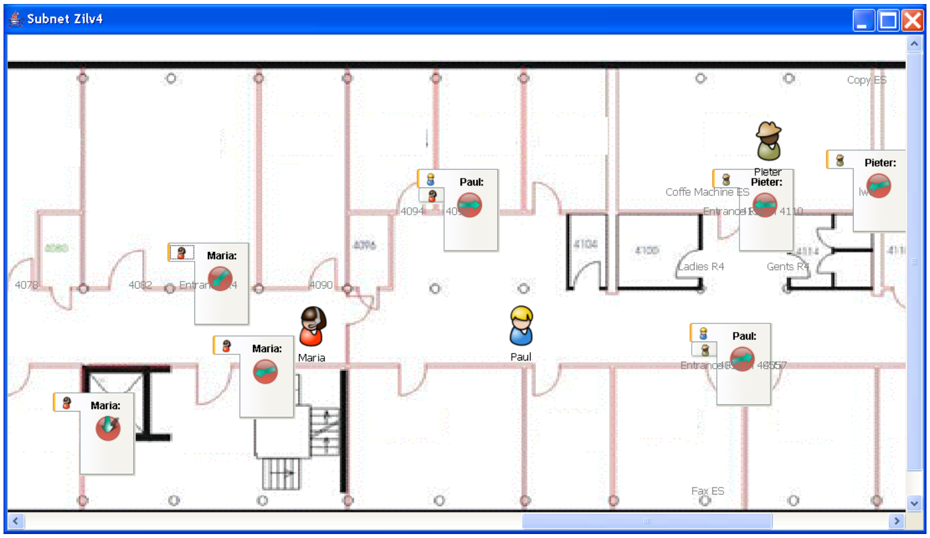

Fig. 4. Example of guidance service in the simulation tool

Presentation Service in the Smart Signs. For privacy reasons, users can provide an alias for the request, which will be shown on the Smart Signs for providing directions.

Apart from a general *Table of Guidance Requests*, the Guidance Service keeps for each Smart Sign the following information:

– *Default Routing Table:* This is a compressed routing table computed for a venue specific *Default User*. On the Smart Signs this table will be used for all users who do not require special directions towards their destination.
– *Table of Special Directions:* This table contains entries for the users or groups who should be provided a direction different than the one obtained from the default routing table in order to reach their destination. An entry is as follows: $[RequestID, UserID/GroupID, Direction]$

When a guidance request arrives, the Guidance Service asks the *Route Planner* to compute *compressed routing tables* for the user preferences. For each Smart Sign it then compares the directions in this table with the respective *default routing table*. If the directions differ, it creates an entry for this request and user ID in the *Table of Special Directions*. If all the users in a group request get the same special directions, then only one entry is created for that group.

Figure 4 shows a simulation of Smart Signs for guidance. Maria and Paul are on the fourth floor and need to go to Room 3126 on the third floor. Paul gets directions to go through the shortest path, which involves walking down the stairs at the right end of the corridor. Maria is directed towards the lift, because she is on clutches. Pieter is inspecting the security installations and is currently being directed to the fire exit at the left end of the corridor.

The service subscribes to the context information of the users (if available) and also receives an event from the Route Planner when the environmental context changes. When the context of a user changes, the data related to that user's request is updated. When the environmental context changes, the service updates the *default routing table* and checks for each user with an active request if the directions for him should change.

When a request's time-to-live expires the Guidance service deletes the request and all the information related to it. It is not necessary to send a remove command to the Smart Signs, because they have the same business rule to remove guidance requests.

3.4 Guidance Presentation Service

Each Smart Sign decides autonomously how to guide the people that it identifies in its vicinity using the information provided by the Guidance Service. An important design decision for the Smart Signs system is to use the location of the user locally on the Smart Signs, so that the users are not tracked. In order to optimize computation and save energy, the Smart Signs maintain a *table of heard tags*, which is used to determine when a tag enters or leaves the vicinity of the sign. The corresponding guidance will be presented as long as the tag is heard.

Periodically, the Smart Sign will clean up the table of heard tags to remove all those signs that it has not heard for a period of time longer than a given threshold, and no records are kept that the user was ever seen. The Smart Sign also cleans up the other tables corresponding to requests, which have exceeded their time to live.

The *Guidance Presentation Service* uses the following tables, which are provided by the *Guidance Service*:

- *Table of Guidance Requests*: The entries on this table indicate the destination of the users and how long the request should be kept active. An entry has the form $[RequestID, TagID/GroupID, Destination, TTL]$. This table is identical in all nodes.
- *Table of Aliases*: This table provides a mapping from user tags to user aliases to show guidance. An entry has the form: $[TagID, Alias]$. This table is identical in all nodes.
- *Mapping of Groups to Tags*: It provides a mapping from group identifiers to set of user tags. Each tag belongs at most to one group. This table is identical in all nodes.
- *Default Routing Table*: This is a compressed routing table that indicates how people are routed if there are no special directions for them. An entry has the form $[Address, Mask, Direction]$. This table is node specific.
- *Table of Special Directions*: This table contains information on how to guide people who should not follow the default direction at that node. An entry has the form $[RequestID, TagID/GroupID, Direction]$. This table is node specific.

Algorithm 1 ShowGuidance(TagID)

```
GuidanceRequest ← TableGuidanceRequests.get(TagID)
if GuidanceRequest = null then
   GroupID ← MappingGroupsToTagsGuidance.getGroupFor(TagID)
   if GroupID ≠ null then
      GuidanceRequest ← TableGuidanceRequests.get(GroupID)
   end if
end if
if GuidanceRequest ≠ null then
   SpecialDirections ← TableSpecialDirections.get(GuidanceRequest.Id, TagID)
   if SpecialDirections ≠ null then
      show SpecialDirections
   else
      show DefaultRoutingTable.getGuidanceTowards(GuidanceRequest.Destination)
   end if
end if
```

Data that is identical on all nodes is broadcast. All other data is uni-cast to individual Smart Signs. Broadcasting is important because it reduces the amount of information that needs to be sent through the WSN. Communication is the most energy-consuming part of a wireless sensor node. Thus, if communication is kept low, we can prolong the time that a Smart Sign can function without a battery change.

Algorithm 1 is called when the node detects a new tag in the vicinity. The algorithm considers guidance requests for the individual and the group to which the individual belongs. If there is a guidance request for the individual, it takes precedence and the one for the group is ignored.

We are currently analyzing how to show the data to the users in a useful and friendly way, especially when many users are in the vicinity of a sign. Our current interface cycles over the data that needs to be shown.

4 Supporting Multiple Interfaces

The Smart Signs system can also be used on a handheld device as *Virtual Smart Signs*. We have implemented such an interface in combination with FLAVOUR [22]. FLAVOUR is a personal application that determines the location of the user using the existing WLAN infrastructure without the need of a centralized system. FLAVOUR does not track people and gives users control over who they share their location information with and under which conditions. The user interface consists of an SVG viewer where the user can view his location, the location of his buddies, and the directions and messages provided by Smart Signs.

The user can run the Smart Signs services locally on his handheld or in a server in the infrastructure that he controls, and thus, keep total privacy. The Guidance and Messaging services are in this case light-weight versions that do not need to deal with multiple users. Virtual Smart Signs can be used even when the venue does not provide Smart Signs in the venue. Additionally, the user can simultaneously use FLAVOUR and the venue's Smart Signs.

The Virtual Smart Signs can also show location-aware messages. On its simplest (non-cooperative) form, the messages are just context-aware annotations of the user. However, the messages can also be obtained from the venue Smart Signs system or from other Virtual Smart Signs, by filtering the messages which are relevant to the user. By relevant we mean that they are sent to the user either personally or to one of the groups he belongs to. To avoid spamming, the user can specify filters on the senders of the messages.

Another possible combination between *virtual* and venue Smart Signs is to use the signs as *ubiquitous displays*. In this case the computing power is on the handheld devices and when available they use the Smart Signs in the venue to display directions and messages for the user.

As was discussed in Section 2.2, there are many situations in which using alternative interfaces, such as haptic interfaces may be desirable. To support other type of interfaces the Smart Signs can act as beacons announcing directions and messages for the nearby users. Also people without specialized hardware, but carrying a PDA or mobile phone could make use of this feature by letting their personal device display the directions, for example in the form of a compass.

The *route planner service* can also be used by a *web route planner* for a venue similar to the normal route planners on Internet (e.g. map24 [19]). The *web route planner* shows the path to follow on a map and provides a text description of the route. The users do not need to be registered in the system, as they can provide their preferences when making the request.

5 Privacy

Following Langhereich's guidelines [15] we have made privacy an important design issue of our architecture and not considered it as an afterthought as in most Ubiquitous Computing projects [16].

As a first step to protect user's privacy, we secured the communication links [17] to prevent eavesdropping. As a second step we designed a system that is privacy aware.

A cornerstone of our design is that the Smart Signs system does not track the users. Each Smart Sign decides locally how to provide guidance and messaging, and after the user leaves the signs vicinity it forgets it has ever heard his tag. Neither is the location of the users communicated to a central server, nor is it stored locally longer than strictly necessary. All guidance requests and messages are deleted from the system as soon as they reach their time to live.

Another important design decision to protect privacy is to prevent that a user's context can be deduced from making requests to the Smart Signs system. As a result, we do not allow user-related context to be used as access rights. For instance such appealing restrictions as 'when the door is open' or 'when an owner is present' would allow a spy to track when the target is in the office based on a simple guidance request.

Furthermore, guidance request provide layers of anonymity: Smart Signs do not show the final destination of a guidance request but only the next step of the way, the user is addressed with a per-request alias, and guidance requests can be posted altogether anonymously with only limited loss of functionality (dynamic changes in user context cannot be considered).

Our privacy claims rely on our trust in the Smart Signs and the wireless sensor network—but, ultimately, this trust can only be achieved with tamper-aware hardware, which our current platform does not provide.

However, a simple modification can limit the amount of information a rogue node can gather regarding the final destination of users. Instead of broadcasting the final address of a request to all nodes, each node receives only the respective next step on the way to the destination. In general, the cost of this additional level of security and privacy is the energy consumption of multiple uni-casts instead of a broadcast, which is significant for WSNs.

It is clear, though, that the Smart Signs (with displays on the nodes) should not be used to convey privacy-critical messages, because everyone near the Smart Signs can see the messages. Only a reliable detection that a person is alone in front of a Smart Sign could allow private message presentation.

Although we safeguard privacy, we have to make a user study to see, whether the potential users of the system really believe in our statements about privacy. In case they do not, the users can always switch off their tags and just switch them on when they have pending guidance requests or expect to see relevant messages in the environment.

6 Conclusions and Future Work

We present a novel use of wireless sensor networks for ubiquitous computing. Traditionally, wireless sensor networks have been used as a source to gather environmental context. However, their capability to execute simple, dynamically changeable business rules make them an ideal platform for context-aware applications, not only for context gathering but also for actuating accordingly. In addition, their capacity to create a self configuring multi-hop network make them easy to deploy, even for temporary purposes.

At present, we use over 20 Smart Signs running an early version of the system presented in this paper on a daily basis to present messages. The displays we are using are too small to be read from a distance, therefore, we are currently trying to find an alternative that improves usability and is also energy efficient. We are also experimenting with haptic interfaces.

The Smart Signs Service with the Guidance Service, Route Planer, and World Model are up and running. We are using these together with the Deployment Tool and the Simulator to simulate different venues. After experimenting with early versions of the messaging and guidance presentation services, we are now implementing them as described in the paper.

The Emergency Service is still an early prototype. With it we will extend the use of the context infrastructure and experiment with rule engines for context reasoning and business rules.

We envision many services that can make use of the Smart Signs. For example a *calendar service* could obtain information from the user's calendar about meetings within the venue that the user needs to attend and make guidance requests on his behalf. This service could also generate automatic messages on the user's door to indicate when the user will be back or what activity the user is performing (e.g., in a meeting, in a workshop until Friday) that should only be shown to certain groups of people.

References

1. Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3(5):421–433, September 1997.
2. Alice Angus and Giles Lane. Urban tapestries: public authoring in the wireless city contexts filmscript. Video in Pervasive'04., April 2004.
3. Ariadne.
4. J. Baus, A. Krüger, and W. Wahlster. A resource-adaptive pedestrian navigation system. In *Proc. of the 2002 Int. Conference on Intelligent User Interfaces (IUI'02)*, pages 15–22, San Francisco, California, USA, 13–16 January 2002. ACM.
5. Christian Bornträger, Keith Cheverst, Nigel Davies, Alan J. Dix, Adrian Friday, and Jochen Seitz. Experiments with multi-modal interfaces in a context-aware city guide. In *Mobile HCI*, volume 2795 of *LNCS*, pages 116–130. Springer, 2003.
6. S. Bosman, B. Groenendaal, J. W. Findlater, T. Visser, M. de Graaf, and Panos Markopoulos. Gentleguide: An exploration of haptic output for indoors pedestrian guidance. In Luca Chittaro, editor, *Mobile HCI*, volume 2795 of *LNCS*, pages 358–362. Springer, 2003.
7. P. J. Brown. The stick-e document: a framework for creating context-aware applications. In *Proc. of EP'96, Palo Alto*, pages 259–272, January 1996.
8. K. Cheverst, A. Dix, D. Fitton, C. Kray, M. Rouncefield, C. Sas, G. Saslis-Lagoudakis, and J. G. Sheridan. Exploring bluetooth based mobile phone interaction with the hermes photo display. In *Proc. of Mobile CHI*, pages 47–54. ACM, 2005.
9. D. Chincholle, M. Goldstein, M. Nyberg, and M. Eriksson. Lost or found? a usability evaluation of a mobile navigation and location based service. In F. Patern, editor, *Proc. of Mobile HCI*, volume 2411 of *LNCS*, pages 211–224. Springer, 2002.
10. Carmine Ciavarella and Fabio Paternò. Design criteria for location-aware, indoor, pda applications. In *Mobile HCI*, volume 2795 of *LNCS*, pages 131–144. Springer, 2003.
11. Vlad Coroama. The chatty environment – a world explorer for the visually impaired. In *Adjunct Proc. of Ubicomp 2003*, Seattle, Washington, oct 2003.
12. Cricket.
13. Jan V. Van Erp. Presenting directions with a vibrotactile torso display. *Ergonomics*, 48(3):302–313, 2005.
14. Giovanni Iachello, Ian E. Smith, Sunny Consolvo, Gregory D. Abowd, Jeff Hughes, James Howard, Fred Potter, James Scott, Timothy Sohn, Jeffrey Hightower, and

Anthony LaMarca. Control, deception, and communication: Evaluating the deployment of a location-enhanced messaging service. In *Ubicomp*, pages 213–231, 2005.
15. M. Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In *Proc. of Ubicomp 2001*, Atlanta, GA., 2001.
16. Marc Langheinrich and Saadi Lahlou. Troubadour approach to privacy. Technical Report Disappearing Computer Initiative 15.3.1, Ambient Agoras, November 2003.
17. Y. W. Law. *Key management and link-layer security of wireless sensor networks : Energy-efficient attack and defense.* PhD thesis, CTIT Ph.D.-thesis Series 05-75, Univ. of Twente, December 2005.
18. M. Mankins. Location linked information: A framework for emergent, location-based content deployment. Master's thesis, Massachusetts Institute of Technology, 2003.
19. Map24.
20. N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. In *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing*, volume 1927 of *LNCS*, pages 157–171, Bristol, England, January 2000. Springer.
21. Allen Ka Lun Miu. Design and implementation of an indoor mobile navigation system. Master's thesis, Massachusetts Institute of Technology, May 2002.
22. K. Muthukrishnan, N. Meratnia, M. E. Lijding, G. Koprinkov, and P.J.M. Havinga. WLAN location sharing through a privacy observant architecture. In 1^{st} *Int. Conf. on Communication System Software and Middleware*, page to appear, New Delhi, India, Jan 2006. IEEE.
23. J. Nichols, J. Wobbrock, D. Gergle, and J. Forlizzi. Mediator and medium: doors as interruption gateways and aesthetic displays. In *Proc. of DIS*, pages 379–386. ACM, 2002.
24. Jon Postel. Internet protocol. Technical Report RFC 791, IETF, September 1981.
25. David Ross and Bruce Blasch. Wearable interfaces for orientation and wayfinding. In *Proc. of the* 4^{th} *Int. ACM Conference on Assistive Technologies (ASSETS-00)*, pages 193–200. ACM, nov 2000.
26. Enrico Rukzio, Albrecht Schmidt, and Antonio Krüger. The rotating compass: a novel interaction technique for mobile navigation. In *CCHI '05 extended abstracts on Human factors in computing systems*, pages 1761–1764. ACM Press, 2005.
27. Timothy Sohn, Kevin A. Li, Gunny Lee, Ian E. Smith, James Scott, and William G. Griswold. Place-its: A study of location-based reminders on mobile phones. In *Ubicomp*, pages 232–250, 2005.
28. J.S. Zelek, S. Bromley, D. Asmar, and D. Thompson. A haptic glove for tactile-vision sensory substitution for wayfinding. *Journal of Visual Impairment and Blindness, special issue on technology*, 97(10):621–632, October 2003.