# **Seamless Resource-Adaptive Navigation**

Tim Schwartz, Christoph Stahl, Jörg Baus, and Wolfgang Wahlster

# **1** Introduction

Research in the project RENA (REsource-Adapative NAvigation) together with DFKI GmbH, BMW Research and Technology AG, and Eyeled GmbH has been concerned with the conceptual and methodological foundations and the design of a resource-adaptive platform for seamless outdoor and indoor navigation that can serve as a basis for product development by the companies in the RENA consortium. Future in-car assistance systems will have a user interface, which adapts to the driver's current exposure caused by the actual traffic situation. Based on concepts developed in [15] such in-car assistance systems will use the car's serial-production sensory equipment to detect the driver's momentary cognitive load without the additional use of biosensors attached to the driver. In case the system detects that the driver has high cognitive load or is in stress, system messages, which are not timecritical, like directions from the navigation system or incoming telephone calls will be delayed until the driver's cognitive load decreases. Besides these new features, in-car assistance systems will use car-2-car and car-2-X communication capabilities to exchange information about the traffic situation and road conditions with other cars and their drivers. Such information gained from the car's sensory equipment, e.g., rain and moisture sensors, sensors from the ABS, and/or sensors from airbags will be mapped onto concepts defined in local danger warning ontologies to realize local danger warnings (see [5, 8]). In parking garages, where the car's assistance system cannot receive GPS signals, the car's position is determined using the car's build-in gyroscope as described in [18] or on the basis of active RFID positioning (see [12]).

The distinguishing feature of RENA's ubiquitous navigation service is its adaptability to the user, the situation, and the technical resources. Various positioning and wireless communication technologies are combined synergistically within the developed platform. Due to the fact that those aforementioned theses are still under nondisclosure agreement up to the time of preparing this chapter, we will focus

T. Schwartz (⊠)

DFKI GmbH and Saarland University, 66123 Saarbrücken, Germany e-mail: Tim.Schwartz@dfki.de

M.W. Crocker, J. Siekmann (eds.), *Resource-Adaptive Cognitive Processes*, Cognitive Technologies, DOI 10.1007/978-3-540-89408-7\_11, © Springer-Verlag Berlin Heidelberg 2010

on YAMAMOTO (Yet Another MAp MOdeling TOol), a map modeling toolkit for indoor pedestrian navigation, and our positioning system LORIOT (Localization and ORientation for Indoor and Outdoor EnvironmenTs). Both systems serve as the core components for different applications such as route visualization and a location-based To-Do Organizer.

The remainder of the chapter is structured as follows: In the next section we will review two navigation systems, which serve as starting points for the later developments. Then, we will give a short overview on the overall system's architecture followed by detailed descriptions of the core components YAMAMOTO and LORIOT. The next section will introduce applications that use both components in order to help users solve different tasks in the navigation domain. The last section will summarize the results.

#### 2 REAL and BPN as the Basis of our Extensions

In this section we provide a brief review of two navigation systems which were developed within our project and served as the basis for further developments, which are described in the remainder of the chapter.

The REAL system was a hybrid pedestrian navigation aid that could tailor its graphical route descriptions to limited technical resources of different devices and the user's limited cognitive resources. Based on the introduced concept of active and passive location sensitivity, a navigation aid for pedestrians was developed. Active location sensitivity delegates the computational burden to the mobile device. The mobile device actively detects the actual location on its own. According to this information the presentation of graphical route descriptions is generated automatically to comply with the user's requests. In contrast to this, passive location sensitivity reallocates nearly all necessary computations to the instrumented environment. The mobile device just passively presents the information that it receives from senders in its local environment, specifically information prepared for the actual location. The navigation aid integrated three subsystems: (i) a stationary information kiosk, (ii) IRREAL (Infrared REAL) for indoor navigation tasks based on infrared senders, and (iii) ARREAL (Augmented Reality REAL) for outdoor navigation based on GPS-satellites. IRREAL uses passive location sensitivity, whereas ARREAL relies on the active counterpart. The whole system has been designed in such a way that the changeover between both adaptation paradigms is barely noticeable for the user. For a more detailed look on the system please refer to [3, 4].

In cooperation with BMW AG, the BMW Personal Navigator (BPN) was developed. The system combines a desktop event and route planer, a car navigation system, and a multi-modal, in-, and outdoor pedestrian navigation system for a PDA. BPN offers a situated personalized navigation service and seamlessly integrates 2D and 3D maps with speech in- and output on the mobile device. With its research focus on multi-modal interaction, BPN allows the user to interact with the navigational map through the combined use of speech (English and German) and stylus gesture. We investigated three different types of situations in which navigational services may be of interest. We aimed at providing a service that transparently combines the desktop PC at home, a built-in car navigation system, and a PDA. Such a situated personalized navigation service allows travelers to prepare their trips at home in order to obtain route directions and maps, choose personalized events of interest at their destination, book an appropriate hotel, and retrieve further information, like the current weather. This information is collected and stored in a travel itinerary for each trip. With the help of the PDA, travelers can also make use of their travel itinerary in the car and as pedestrians on-foot, both inside and outside of buildings. At home the desktop PC is used to make all travel arrangements provided by a personal navigation server that can be accessed over the Internet. The travel itinerary is then synchronized with the PDA which then allows access to the travel itinerary without the need for a direct Internet connection. In the car, the PDA connects locally to the car navigation system, which in turn allows to transfer the travel itinerary from the PDA to the car navigation system. During the navigation task in the car, the PDA remains predominantly silent, and the car navigation system takes control in guiding the traveler to the selected destination. Before leaving the car, the PDA receives the actual park spot coordinates, which are added to the travel itinerary and may help to find a way back to the car later. For the pedestrian, the PDA plays a much more vital role. It displays the 3D map information included in the travel itinerary and guides the traveler with verbal and graphical route directions. It can also be used to store geo-referenced user data (e.g., voice memos) and respond to multi-modal requests regarding landmarks in the environment as described in [26, 14].

# **3** Overall System Architecture of the New Navigation Framework

Based on the results of the aforementioned prototypes, we will now introduce the components of the new navigation framework and the way they interact with each other as illustrated in Fig. 1. The complete navigation framework consists of different services which run on workstations, servers, and mobile devices. Furthermore, our lab is instrumented with active RFID tags and infrared beacons for positioning purposes, and Bluetooth-enabled public displays for location and user adaptive information services. In more detail, the following base applications comprise the software environment:

- *YAMAMOTO* is an application that was developed for easy and rapid modeling of indoor and outdoor environments. YAMAMOTO is also able to calculate routes from one room to another in one building or from a room in one building to a room in another building.
- *LORIOT* is an always best positioned (ABP) system which uses active RFID tags and infrared beacons as well as GPS to calculate the position of users in indoor and outdoor environments. It runs on PDAs and has the highest precision of all positioning systems described in this chapter. A user can choose to reveal their position to the instrumented environment or protect their privacy.



Fig. 1 Overview of all components discussed in this chapter, and how they are interconnected

- *UbisWorld* is a ubiquitous user modeling service that provides user models as well as spatial ontologies. An important feature of UbisWorld is the socalled UbisNames which provide a unique symbolic identifier for each entity (people, places, objects) that is modeled by the service. A more detailed description of UbisWorld can be found in [24] in this volume.
- *iROS EventHeap* The iROS (Interactive Room Software Infrastructure) Event-Heap [13] is a part of the Stanford Interactive Workspaces Project. We use the EventHeap to interconnect the different services and applications that either run on servers, workstations, PDAs, or mobile phones. Each application or service can register at the EventHeap and can then post and receive messages (events).

In the following we will provide a detailed view on the core components YAMAMOTO in Sect. 4 and LORIOT in Sect. 5.

## **4** Providing Map Material for Pedestrian Navigation

Our research aims toward a system that provides ubiquitous navigational aid for its users, with emphasis on indoor environments, but which also covers outdoor places and routes on a large scale. As mentioned in Sect. 2 this vision of a seamless indoor–outdoor navigation system has already been implemented in the REAL and BPN research prototypes (see [3, 14, 4]). Whereas these prototypes have been quite successful in demonstrating the overall concept, we have learned several lessons about location modeling. The BPN system is based on a commercial street database, which has been imported into an open source GIS system and manually extended by indoor route segments. This approach lacks the scalability that would be required for a truly ubiquitous location model, as too many buildings need to be represented and maintained in a single database.

Since the environment model is solely based on a path network which is represented as a graph of line segments in two dimensions, it has severe limitations for the pedestrian navigation domain. The situation of a pedestrian differs from driving tasks, since the user is not bound to follow paths or streets. Instead users typically cross open spaces directly following their line of sight. The model should particularly reflect this and represent places as polygonal regions.

Furthermore, since the GIS system's routing module operates with 2D geographic coordinates, it takes several workarounds to denote destinations in the upper floors of buildings. Inside the building, no GPS data can be received, so the BPN system used infrared beacons, and the position of the user was looked up from a database that returned the geometric coordinates of the received beacon IDs (16 bit of information). The installation of the beacon infrastructure took several days, since no tools were available at this time to graphically model the position, range, and orientation of the beacons. Other difficulties arose from the fact that three different location models were required. Start and destination addresses are usually given as geographic locations (postal addresses) and have to be mapped to physical locations in the WGS84 [7] coordinate system (longitude, latitude) used by GPS. These geographical coordinates have to be mapped to screen coordinates (x, y)in the map's texture bitmap reference system in order to visualize the position of the user. Indoors, no WGS84 coordinates are known, and the paths and beacons have been entirely modeled in bitmap coordinates instead. The alignment of the indoor space with the outdoor space has been done manually and hence was error prone. Besides map visualization, the BPN system has been designed to convey automatically generated verbose instructions to the user, such as "turn right after 10 m." As the underlying location model consisted of a set of two-dimensional floor maps without height information, additional annotation workarounds were required to guide the user through a staircase, e.g., "Please go up the stairs to the 2nd floor."

The commercial providers of navigational maps for mobile systems have recognized the benefit of 3D visualizations, but they are still focused on outdoor environments. As pedestrians spend most of their time inside buildings, indoor environments need to be modeled in 3D with multiple floors and landmarks. Indoor's decision points are more complex than outdoor's because stairs and elevators add choices. For the same reason, routes cannot be depicted easily in a single map, so that indoor wayfinding tasks generally pose a high cognitive load to the user.

In summary, we conclude the following research issues for pedestrian navigation in mixed indoor/outdoor environments from our previous experiences with our research prototypes:

- scalability and maintainability of the underlying location model
- polygonal representation of regions instead of abstract line segments
- mapping of beacons for indoor positioning
- hybrid geographic (symbolic) and physical (geometric) location modeling
- modeling height information.

During the course of our project, we have continuously developed our own map modeling toolkit, which is called YAMAMOTO. The toolkit is positioned between proprietary 2D location models that are typical for ubiquitous and pervasive computing research projects on indoor navigation and professional 3D CAD (computeraided design) tools for architects. CAD tools require a high level of experience; the designer has to manually cut out windows and doors from solid walls and take care about window sills, choices of door handles, or steps of a staircase. Such a high level of detail is not required for route finding and presentation purposes. Our approach strives to minimize the modeling effort. By following the motto to keep everything as simple as possible, we have intentionally reduced the degrees of freedom by half of a dimension in order to allow for a simpler and easier to learn user interface.

Now what exactly does this mean, and what are the implications? Rooms of a building are represented only by their outline as flat polygon objects. Each polygon object is defined by an ordered sequence of vertices. Each vertex is represented through Cartesian coordinates as a triple of (x, y, z) values. The z-value allows representing the room's height above ground level, so that multiple floors can be represented. Polygons can have several symbolic attributes, such as name, type, and accessibility for pedestrians. Polygons that are defined by vertices from two different levels represent connections such as ramps, stairs, or escalators. Figure 2 (left) shows an example, where the polygon "Corridor.14021" is defined as sequence of vertices with index (1, 2, 3, 4, 5, 6, 7, 8). In order to allow for route finding, it is important to know the semantics of connections between polygons. Thus each edge is attributed by their passability: edges that represent walls or windows are set to be "not passable"; in our example, edge (8, 1) represents a wall and edge (6, 7)connects the corridor with the adjacent staircase and is annotated to be "passable for pedestrians." On the right-hand side in Fig. 2, a sample path is shown that has been calculated based on start and end points within the 2.5D location model.

Based on the outlines of the rooms and some additional annotation of type and height, YAMAMOTO automatically creates the building structure in full 3D. By using the predefined building blocks shown in Fig. 3, edges can be visualized in the perspective views as walls, doors, murals, or handrails.

In addition to polygon objects, the mapping of navigational fixpoints, such as beacons or landmarks, is required for the indoor positioning system. For this reason we allow additional geometric primitives, such as points, spheres, and sections, which can be used to represent the position of the beacon or more precisely the reception range of the signal emitted by the beacon.

In YAMAMOTO one can choose among different viewpoints at any time. The orthogonal view shows a top-down projection of the model similar to traditional maps. The perspective view shows the model from an allocentric viewpoint outside



Fig. 2 The 2.5D data model (left) and a route between two points (right)

the model, as shown in Fig. 5, and allows for free rotation and zoom. The user itself can be virtually represented in the model through an avatar object. The egocentric perspective shows the model from the viewpoint of this avatar, see Fig. 5. It allows for the virtual exploration of the modeled environment. It also creates a demand for interior items that could serve as landmark objects for route descriptions. Rooms can be equipped with predefined 3D objects, like shelves, tables, or pictures, as depicted in Figs. 4 and 5.



Fig. 3 Set of building blocks used to automatically create 3D geometry from the 2.5D model



Fig. 4 Allocentric perspective



Fig. 5 Egocentric perspective

Furthermore, the toolkit allows to test the instrumentation of the environment with pervasive computing artifacts, i.e., beacons used for indoor positioning/navigation and public displays. The avatar view lets the designer virtually examine the visibility of the displays from various positions and helps to identify the best configuration, as described in [22]. In particular the possible interpretation of graphical signage, e.g., an arrow pointing upward, can be ambiguous depending on the actual context of the building. Such situations can be virtually evaluated and resolved before the signs are deployed.

As the toolkit has been designed with pedestrian navigation in mind, it includes a route finding module. It is able to generate routes between any two points in a model, which follow the line-of-sight whenever possible instead of a restrictive path network. Since even multi-story buildings can be represented as a single mesh, the pedestrians will be routed through staircases, if needed. During the modeling process, the results of the route finding module can be tested at any time within the editor.

#### 5 The Always Best Positioned Paradigm

Today's navigation systems are designed to work for a specific platform in a welldefined environment, but they usually fail to work when the situation of the user is dynamically changing, e.g., when the user has to combine different means of transportation to reach a destination. One essential issue in this context is the switch between different positioning technologies. A navigation system intended to support its mobile users has to seamlessly cope with this technical problem and must be able to adapt to different technical constraints.

#### 5.1 Exocentric and Egocentric Localization

Robust and resource-adaptive positioning is critical for the success of pedestrian navigation services. If the user is staying outside, localization can be done with the use of GPS. Unfortunately, GPS is not working in indoor environments, e.g., a shopping mall or an airport. There have been numerous attempts to overcome this restriction (e.g., [2, 25, 9]). All of these systems use some sort of senders (ultrasound beacons, infrared beacons, WiFi-hotspots, RFID tags, Bluetooth beacons, to name but a few) and corresponding sensors to detect or read these senders. Basically there are two options to set up such a system: installing sensors in the building and letting the user wear the sender or installing the senders in the environment and letting the user wear the sensors.

In the former case, the so-called exocentric localization (see left side of Fig. 6), the user is sending information to the environment and some centralized server uses these data to calculate their position. In other words, the user is tracked. In the latter



Fig. 6 Exocentric localization (left): the dataflow is from the user to the environment. Egocentric localization (right): the dataflow is from the environment to the user

case, the egocentric localization (see right side of Fig. 6), the user receives information from the environment and their personal device uses the data to calculate the current position.

# 5.2 LORIOT

In [6] and [21] we describe the basics of an egocentric localization system that uses Geo Referenced Dynamic Bayesian Networks (geoDBNs) to determine the position and orientation of a user.

Based on this technique we developed a seamless indoor/outdoor positioning system named LORIOT (Location and ORientation in Indoor and Outdoor environmenTs), which uses infrared beacons (IR beacons) and active RFID tags to determine the user's position in indoor environments and a GPS receiver for outdoor localization. LORIOT runs on Windows Mobile PDAs and uses the built-in infrared receiver, an attached active RFID reader, and a Bluetooth GPS receiver as sensors.

For indoor positioning the environment has to be instrumented with IR beacons and/or active RFID tags. These beacons and tags can be installed at the ceiling of the building and act as electronic landmarks. The reason to use two kinds of senders is that both technologies differ in their features, precision, and cost: IR beacons send out a 16 bit wide identification code. Due to the physical attributes of light, receiving such a beacon gives a very high probability that the user is standing near that particular beacon. Furthermore, if we know the direction of the infrared light beam, we can determine the user's direction. However, the disadvantage of IR beacons is that an IR sensor must be in the line of sight of the beacon and can thus be very easily blocked.

On the other hand, RFID tags send their information via radio waves, which can be received even when the PDA resides in the user's pocket. Due to reflections and damping of radio waves, receiving an RFID tag gives only little evidence that the user is standing in its vicinity and the signal does not contain any directional information. Besides a unique, hardwired identification code, each active RFID tag contains a 56 byte wide, free accessible memory which we use to store its own geo-coordinates. These geo-coordinates, in the format of latitude, longitude, and elevation based on WGS 84 [7], are obtained with the help of our map modeling tool YAMAMOTO (see Sect. 4).

By combining different sensor readings with the help of dynamic Bayesian Networks, we achieve a positioning system that follows the above mentioned Always Best Positioned paradigm: As long as there are either IR beacons, RFID tags, or GPS satellites detectable, we will be able to estimate a position whose precision depends on the type of the sender. If we can receive all, we will get an even higher precision. The system adapts to the available resources (senders as IR beacons, RFID tags, and GPS satellites, or attached sensors as an infrared port, an active RFID reader card, or a GPS receiver) that can be used to find out about its own position.

In the following, we describe the idea of Geo Referenced Dynamic Bayesian networks and how they are used to accomplish a sensor fusion and cancel out false readings.

Bayesian networks (BNs) are a computational framework for the representation and the inference of uncertain knowledge. A BN can be represented graphically as a directed acyclic graph (see Fig. 7). The nodes of the graph represent probability variables. The edges joining the nodes represent the dependencies among them. For each node, a conditional probability table (CPT) quantifies these dependencies.

Dynamic Bayesian networks (DBNs) are an extension of Bayesian networks. With a DBN, it is possible to model dynamic processes: Each time the DBN receives new evidence a new time slice is added to the existing DBN. In principle, DBNs can be evaluated with the same inference procedures as normal BNs; but their dynamic nature places heavy demands on computation time and memory. To cut down these computational costs, it is necessary to apply so-called roll-up procedures that cut off old time slices without eliminating their influence on the newer time slices.

#### 5.2.1 Estimation of the User Position

For the purpose of positioning, we let such a DBN represent the characteristics of the used senders. Figure 7 shows two geoDBNs: The one on the left was designed for indoor positioning only and thus contains only nodes for the IR sensor and the



Fig. 7 *Right*: a simple geoDBN with IR- and RFID-Sensor nodes. *Left*: an extended geoDBN with additional GPS sensor node

RFID sensor. The geoDBN on the right is an extension, which also includes a node for a GPS sensor (GPS receiver) to enable seamless indoor/outdoor positioning. It is important to note that we do not use the DBN to represent all the senders that are physically installed in the environment. We use one small DBN that prototypically describes the reliability of the sender types (assuming that all senders of a certain type have the same reliability). This prototypical DBN is instantiated several times during the runtime of the system and each instantiation gets assigned to a geo coordinate *GeoPos*.

Figure 8 shows the network (with time slices) that we use in LORIOT. The top node at time slice *t* (labeled UserPos=GeoPos?) represents the probability that the user is standing at the assigned geo-coordinate *GeoPos*. The node to the left of it (UserPos=GeoPos?\_1) represents the probability that was calculated in the previous time slice t - 1. The bottom nodes (IRSensor, RFIDSensor, and GPSSensor) represent the probability that an IR beacon and/or an RFID tag installed at *GeoPos* can be detected under the condition that the user is standing at *GeoPos* and/or that the GPS receiver reports the coordinates of *GeoPos*, respectively.

Receiving an infrared signal gives very high evidence that the user is standing near the respective beacon (the infrared sensory data is nearly noise-free). Receiving an RFID tag gives smaller evidence that the user is standing near the tag, since the reader has a reading range of about 10 m and due to reflections of the radio waves, some RFID tags that are far away from the user can be detected. Modern GPS receivers are able to receive GPS signals within buildings if they are near to windows or outer walls, but the accuracy of the received GPS position is very low. Figure 9 shows the received GPS coordinates of a GPS receiver resting on a windowsill over a period of 25 min. In this set of data, the maximum deviation from the actual position is 444 m. Because of this high variation, the probability that the reported GPS position really coincides with the user's position is also very low.

The different characteristics of each technology are coded in the conditional probability tables (CPTs) of the IRSensor-, RFIDSensor-, and GPSSensor-nodes. The networks with their assigned coordinates are the geo-referenced dynamic Bayesian networks (geoDBNs). Each geoDBN represents the believe that the user is standing at the associated coordinate.

As stated above, the active RFID tags have a small internal memory that can be used to read and write data. We use this memory to store the coordinate of



Fig. 8 A geoDBN with different time slices, which represent the different measurements at sequential points in time



Fig. 9 Deviation of a GPS receiver resting on a windowsill inside a building

the tag. Since IR beacons only send a 16 bit wide identification code, the system needs a database that contains the associations between identification code and the respective geo-coordinate of the beacon. We also store these associations into the RFID tags, together with the information of the lighting direction of the IR beacon, so that the database can be distributed over several RFID tags in the environment. That way, the system can retrieve all the information it needs to calculate its own position from the environment. When a tag, a beacon, or a GPS position is sensed by the PDA, geoDBNs are instantiated and associated with the appropriate coordinates.

The calculation of the user position is done with a weighted sum of the received (or sensed) coordinates, where the probabilities of each geoDBN are used as weights:

UserPos(t) = 
$$\sum_{i=1}^{n} \alpha w$$
(GeoDBN[i]) GeoPos(GeoDBN[i]). (1)

Here *n* is the number of existing geoDBNs at time  $t (n \ge #ReceivedSenders_t)$ , GeoPos(GeoDBN[*i*]) is the coordinate and w(GeoDBN[*i*]) the weight of the *i*th geoDBN.  $\alpha$  is a normalization factor that ensures that the sum of all weights multiplied with  $\alpha$  is 1.

To reduce calculation cost and memory usage the number of instantiated geoDBNs must be as low as possible. To achieve this goal, geoDBNs with a weight lower than *threshold*<sub>use</sub> are marked as unused (these geoDBNs provide only little evidence that the user is in the vicinity of their geo-coordinate). This threshold should match the a priori probability for the geoDBN at its first instantiation. To cope with resource restrictions, a maximum number of possible geoDBNs can be specified. If this number is exceeded, those geoDBNs that provide the lowest estimation will be deleted. To keep the overhead for memory management low or to prevent garbage collection – if the system is implemented in languages like Java or C# – geoDBNs that are marked as unused can be "recycled" by resetting them to initial values and new coordinates.

#### 5.2.2 Orientation Estimation

Besides the information about the position of a user, the information about their walking or seeing direction can also be valuable. A common example is an electronic museum guide that not only needs to know where the visitor is standing but also which exhibit she is currently looking at, so that the system can give the respective explanations. The following describes how the direction information from both sensor types, infrared and active RFID, can be fused together again with the help of a dynamic Bayesian Network.

#### 5.2.3 Orientation Information Through Infrared Beacons

The emitted infrared beam of our IR beacons has a range of about 6 m and a conical transmission characteristic due to the physical attributes of light.

Because of this conical transmission characteristic, the infrared beam is highly directional and the calculation of the walking direction is fairly easy. If the beacon sends its light in direction vector **v** (see Fig. 10) and the user receives the beacon then they are walking in direction  $\mathbf{dir}_{IR} = -\mathbf{v}$ . Of course this is only an estimation since the user can be slightly to the left or right of the main direction **v** due to the conical transmission characteristic. Also note that it is sufficient to use a 2D vector (the projection of the 3D direction vector – that includes the tilt of the beacon – on the *x*–*z*-plane).



**Fig. 10** Part of room plan with direction vectors of two IR beacons

#### 5.2.4 Orientation Information Through Active RFID Tags

The transmission characteristic of active RFID tags is radial. Due to this radial transmission characteristic, the estimation of the direction is not as easy as with the infrared beacons. We estimate the direction as follows: (i) Store the starting position  $P_0$  of the user in a variable *lastPosition*; (ii) With every new calculated position  $P_n$ , calculate the distance to *lastPosition*; (iii) If the distance is large enough (a few meters), calculate the direction vector **dir<sub>diff</sub>** = *lastPosition* –  $P_n$  and store  $P_n$  in *lastPosition*. Repeat the steps (ii) and (iii) with every new calculated position.

#### 5.2.5 Fusion of Orientation Information Through Bayesian Networks

The direction estimation with infrared beacons is rather accurate but the beacons are not always in reach of the user. The direction estimation through difference calculation is always possible (whether there are only RFID tags available or only infrared beacons or both) but it is also inaccurate. A combination of both techniques with a dynamic Bayesian network should give better and more stable results.

In this section we describe how the directional information from the infrared beacons and RFID tags can be combined.

As described in the introduction, we use a dynamic Bayesian network to fuse the direction data. The network itself is rather simple (see Fig. 11): It contains only three nodes and each node contains evidences for direction north, south, east, and west. The topmost node is the user direction node. This is the node that will contain the calculated (combined) direction after the roll-up and inference routines have been calculated.

The lower left node is the node for the infrared-based direction vector. As explained above, the estimated direction  $\mathbf{dir}_{IR}$  is slightly inaccurate because the user can stand to the left or right of the main sending direction **v**. This fact is encoded in the conditional probability table (CPT) of the infrared node: the probability that the user is heading in the estimated direction  $\mathbf{dir}_{IR}$  is set to 0.9, the probability that she has a variation perpendicular to  $\mathbf{dir}_{IR}$  is set to 0.045 (in both directions), and the probability that she is walking backward ( $-\mathbf{dir}_{IR}$ ) is set to 0.01.

The right-hand node is the node for the direction that was calculated through the difference of two points ( $dir_{diff}$ ). Due to the fact that this direction is rather inaccurate, the CPT entries are as follows: 0.65 for heading exactly in direction  $dir_{diff}$ , 0.15 for the variance perpendicular to  $dir_{diff}$ , and 0.05 for walking backward.



**Fig. 11** Dynamic Bayesian network for direction fusion

#### 5.2.6 Decomposition of a Direction Vector into Evidence Values

We need a way to represent an arbitrary direction vector as evidence values, so we can insert the estimated directions  $\mathbf{dir}_{IR}$  and  $\mathbf{dir}_{diff}$  in the respective nodes of the DBN. We do this by decomposing it in its *x*-component *X* and its *y*-component *Y* (see Fig. 12, left) and by making sure that the sum of the derived evidences is 1: If Y > 0 (this means the user has a north component in their direction), the evidence for north is

$$e(north) = \frac{Y^2}{X^2 + Y^2}$$

and the evidence for south e(south) is set to 0. If Y < 0 (the user has a south component in their direction), it is vice versa:

$$e(north) = 0$$
 and  $e(south) = \frac{Y^2}{X^2 + Y^2}$ 

The same principle applies for the x-component: If X > 0 (direction has east component), then

$$e(east) = \frac{X^2}{X^2 + Y^2}$$
 and  $e(west) = 0$ .

If X < 0 (direction has west component), then

$$e(west) = \frac{X^2}{X^2 + Y^2}$$
 and  $e(east) = 0$ .

Note that the sum of the evidences is always

$$\frac{X^2}{X^2 + Y^2} + \frac{Y^2}{X^2 + Y^2} = \frac{X^2 + Y^2}{X^2 + Y^2} = 1.$$



Fig. 12 Decomposition and composition of direction vector

#### 5.2.7 Composition of a Direction Vector out of Evidence Values

The estimated directions  $dir_{IR}$  and  $dir_{diff}$  are decomposed as described above and the resulting evidences are inserted into the lower nodes of the DBN for each time slice, one time slice is added for every new measurement of the localization system. After performing the roll-up and the inference procedures of the DBN, the user direction node will contain evidences for north, south, east, and west components of the new direction. These evidence values must be combined to get a new direction vector. We do this by treating the components as a parallelogram of forces (as known from physics, see Fig. 12, right). The new direction vector  $dir_{res}$  consists of the x-component X = e(east) - e(west) and the y-component Y = e(north) - e(south)of the new calculated evidence values. The length of the calculated direction vector  $dir_{res}$  can be used as a confidence value, e.g., the longer the vector, the higher the confidence that the computed direction is correct.

#### 5.2.8 Example Calculation

Figure 13 shows an example calculation. The top left node shows the result of the previous time slice, above it the composition of the calculated direction vector can be seen. Note that the evidence values for east and west are exactly the same, while the evidence for north is much bigger than for south. This causes the resulting vector point strictly to north. The two nodes below show the new evidences of the current measurement. The infrared direction has 100% evidence for east (the vector can be seen below the node), the difference direction node has the same evidence value for east and south.

The network on the right side of Fig. 13 shows the result after the inference routines have been carried out. The top node has strong evidence for east and still



Fig. 13 Example calculation showing two time slices

little evidence for north. This is because both direction nodes give evidence for east but only the, perhaps inaccurate, difference direction node gives evidence for south. Because the DBN includes the previous calculated direction, the north component is still present in the new direction. This is the expected result of such a system, because it helps to smooth out fast jumping of the direction information and emphasizes the direction for which is has the most evidence.

After the detailed presentation of the core components in the preceding two chapters, we will now take a look at different applications whose services are based on the aforementioned components.

# 6 Implementing a Seamless, Proactive User Interface

In this section we pick up the idea to use animated 3D visualizations of indoor routes as navigational aid for pedestrians instead of static floor plans, which has been introduced earlier in [3] as a result of the SFB 378 project REAL. Visitors were able to choose a destination from a stationary information booth. Depending on their user profile and current stress level, such as time pressure, adaptive presentations have been rendered and were shown on a large screen. On their way, pedestrians were assisted by the mobile system IRREAL which iteratively presented new maps each time the user passed an infrared beacon. The concept of a seamless user interface for pedestrian navigation has been introduced in the BPN project, where a PDA connects three different situations and user interfaces: preparing a business trip at home, driving in the car, and finding the way from the car to the office in the street and in the building (see [14]).

We have reimplemented the functionality of the prior systems based on the current YAMAMOTO toolkit by using its routing and rendering engine, and the new positioning system LORIOT, so that the new systems, HYBNAVI and VISTO, are able to automatically generate arbitrary indoor routes across multiple levels. HYBNAVI is a tool for visualizing generated route descriptions. VISTO (VIdeos for SpaTial Orientation) [20] implements a seamless, proactive user interface based on the UBIDOO [23] task planner and the IPLAYBLUE framework [19] for ambient displays.

#### 6.1 Hybrid Navigation Visualization

We developed a mobile navigation visualization system for pedestrians called HYB-NAVI (HYBrid NAvigation VIsualization). In contrast to the BPN, all calculations, e.g., routes, route descriptions, and route visualization, are done on the mobile device of the user. HYBNAVI uses YAMAMOTO maps for these calculations and is not only able to navigate a user from one room to another in a multi story building, but also from a room in one building to another room in a different building. The system seamlessly switches from indoor to outdoor representations and dynamically re-calculates the route if the user deviates too far from the planned route. Since HYBNAVI uses a VRML engine to render the YAMAMOTO map material, it can offer several 2D and 3D viewing perspectives. Based on psychological findings in [27] three perspectives which serve different purposes have been identified:

- *Immersed View*. This is an egocentric perspective and shows the same view that the user currently has. Orientation and position of the camera are the orientation and position of the user. This view is best for navigation if the user is not required to gain a mental representation of the environment.
- *Tethered View.* In this view the camera is positioned above and behind the user. Its orientation equals the orientation of the user. This view allows the user to gain more information about their surroundings, since it reduces the 'keyhole effect' of the immersed view.
- *Plan View.* This is a 2D bird's eye view, where the camera is placed directly over the user's position. This is most favorable if the user wants to build a mental map of their surroundings.

Figure 14 shows each of the three views with the user standing at the same position. The use of YAMAMOTO map material also enables the system to show detailed objects, like soda machines or tables as seen in Fig. 15 (left and middle).



Fig. 14 Immersed view, tethered view, and plan view



Fig. 15 (*left, middle*) Detailed objects like soda dispensers and tables can act as landmarks. (*right*) HYBNAVI's view while navigating a user

These objects can serve as important landmarks for the user. Besides the rendering of the building interior, HYBNAVI also shows the planed route, directional arrows, a virtual compass, and an hourglass that visualizes the estimated walking time (see Fig. 15, right).

#### 6.2 VISTO: Videos for Spatial Orientation

VISTO has been designed to proactively support arbitrary navigational tasks in indoor environments. The user interface is based on our own, web-based organizer called UBIDOO (UBIquitous To-DO Organizer).

In some public places you can already find information kiosk systems that present basic information about local services and maps for orientation. However, these systems are not frequently used even though they provide interesting information for the user. We believe this is because it takes too many steps of interaction until the desired information can be retrieved through the user interface. Typically, the user has to begin the dialogue with some basic choices, such as language, and navigate through many sub-menus. Often it feels easier to ask people or consult traditional maps, especially in situations of time pressure (e.g., if we arrive at the airport we would not like to experiment with some machine and try to figure out how it works). Yet we believe that public or ambient displays, if properly designed, have several advantages over personal devices. First, the diversity of mobile devices regarding their operating systems and technical capabilities makes it hard to implement assistance applications that run on any PDA or mobile phone. Second, all kinds of connectivity-related issues arise today, like WLAN access keys or different GSM frequency bands worldwide. Finally, mobile devices would require standardized indoor positioning services like LORIOT in order to provide navigational aid, which will not be available in the near future. Hence we use wall-mounted public displays whose position and orientation within the built environment can be statically modeled.

We have designed VISTO as a user-adaptive system which is able to automatically recognize the user's identity from their mobile device through the wide-spread Bluetooth wireless protocol, as the user passes an ambient display. The trick is that no data connection is required, so the users do not have to interact with their mobile device in order to set up anything except enabling Bluetooth. Once the user is identified, VISTO can retrieve information about the current tasks from the web-based UBIDOO organizer. UBIDOO applies Activity Theory in a sense that the user can hierarchically structure their tasks and relate them to a set of predefined activities, which are accessible through the UbisWorld ontology. Vice versa, the spatial location model of UbisWorld allows to associate places with typical activities. The combination of both relations, task-activity and task-location, yields in a powerful association between tasks and places. Thus VISTO can automatically sort out only relevant dates and tasks from the user's organizer and suggest a clear and useful set of destinations in the local area.

#### 6.2.1 The Ubiquitous To-Do Organizer UBIDOO

With the advent of mobile computing devices, personal information management (PIM) tools have been implemented on PDAs and are now a common feature of mobile phones. Usually, the devices offer a calendar that allows to add events and set an alarm time for a reminder. The calendar is typically supplemented by a to-do list or notepad, where the user can add tasks that have no certain date. Whereas the conceptual model behind these PIM tools is well understood by both designers and users, it has obvious drawbacks that we want to address in this section.

Our ubiquitous task planner UBIDOO integrates the conceptual models of a diary and a to-do list into a hierarchical task tree, where each task can have multiple subtasks (see [10, 11]). UBIDOO is running on a web-server and can be accessed everywhere via the Internet. The user interface for desktop PCs is split in three frames, as shown in Fig. 16. On the left-hand side, the users' tasks and subtasks are shown in a foldable tree structure. Each task acts as a reminder to do a certain activity or action. The subtasks can either represent alternative choices, multiple goals, or fixed sequences of actions. Similar to a to-do list, the status of each task is graphically represented by a checkbox. By selecting a task from the tree on the left, a summary of its details (dates, location, and description) is shown and icons allow moving or deleting this task. In the frame to the right, various properties of the selected task can be edited which are arranged into six tabs. The bottom frame provides links to the user profile and a traditional calendar view. The user can also manually choose a location for an adapted task view ("here and now") that will be described later in more detail. For mobile devices, a reduced interface without frames and tabs is also available.

Usually, calendars in mobile devices offer an alarm function that reminds the user on events at a fixed time and date. Setting the proper time for an alarm requires the user to consider everything that needs to be done and prepared before the event, most importantly how to go there. Some events might require the user to go home first and dress accordingly or pickup some things, which takes additional time. The user has to plan and foresee the whole situation under uncertainty. However, often the situation changes in an unpredictable manner and we will not be where we have planned to be. Thus the alarm will remind us too early, unnecessarily distracting us from our work, or worse, remind us too late, and thus we cannot reach the event timely. Our ubiquitous task planner addresses this issue through an adaptive reminder, which continuously recalculates the best time for the alarm based on the current geographic distance between the user and the event. In addition, a general preparation time can be specified that will be considered by the reminder. Tasks can be localized by specifying a location from the spatial ontology in UbisWorld (see the "Place" attribute of a task in the "General" tab in Fig. 16). As the location of the user changes, the task planner updates the distance between the user and all tasks using route-planning web services. We use the eRoute service that is provided by PTV AG and return the estimated driving time between two locations. In the future, web services for public transportation and pedestrians could be easily added. The planner would then choose between car and bus based on the user's preferences.

ubidoo	.com the ubiquitou		Subtask	From "Flight to Corfu"	insert subtask
NewTask	to o o o o o o o o	unzer	General Detail	Series date Links & files Invite friends	Constraints
open all   dose all			Description: Check-In	n (GermanWings)	
future tasks     Ubicomp 2007     @ UM 2007 (Use     open all   dose all     @ current tasks     @ Refill the fridg	r <u>Modelling)</u> je		ODate OInterval H (Earliest) Start: H (Latest) Finish: ODuration: 0 de	Begin at         O Dea           22         02         2007         III           ays         0         hours         30         minutes	IT:00
Buy christmas	ra PowerShotS50 ang - Dragon Son		Prep time: c	Jays 0 hours 30 minutes minder: 2007-02-22 16:00:00	
Diplomarbeit (	Oliver		Relace:	Airport Zweibruecken.930155	choose
B MobileHCI 200	07 Deadline		My Contacts:		<u>choose</u>
OLOCA 2007 Write my diss	ertation			Perm	ission : owner
Bight to Corfu	ix i ermanWings) 🔐 🐨 🖓 🖻 🔯		save	save as prototype	delete
→ 22.02.2007, : Airport Zveibr	17:00 uecken.930155	~			
Calendar	Vour cur	rent position			
User profile	PMS6.930156				
Subscription Logout	(You can enter more places in you H	r user profile) ere and Now			

Fig. 16 The Web-based user interface for the ubiquitous task planner UBIDOO

A further shortcoming of mobile organizers is their small display size. Browsing through calendar entries and large to-do lists is a cumbersome procedure. Hence we have implemented the "here-and-now" view in UBIDOO. It filters the user's tasks according to the current situation, which depends on time and location of the user, so that it displays only those tasks which can actually be worked on. The time horizon of this view is limited by the next binding date. The system performs a route calculation for each task to estimate the actual distance and time to get there. UBIDOO also considers the purpose of locations in terms of activities from the ontology such as "shopping for electronics" instead of a certain store, the adaptive view automatically performs a search within UbisWorld's spatial elements and suggests the closest suitable location nearby for this task. Depending on the current time and location of the user, the system might suggest different places for the same task.

Figure 17 shows the same tasks as seen from two different locations. On the left image, we see the adapted view for Saarbrücken. The first task, swim and relax, is associated with the spatial purpose of waterworld and the planner suggests a place called Calypso in Saarbrücken that is located 7 km (or 11 min by car) from the current location of the user (at his office). A click on the R icon opens a route planner that provides a map and driving directions. The second task reminds the user to buy olives and milk. For the first item, a store called Saarbasar is set by the user as

#### Seamless Resource-Adaptive Navigation



Fig. 17 The location-adaptive task view as seen on a mobile device in Saarbrücken (*left*) and Munich (right)

location, whereas for the second item the general activity "shopping.food" has been specified so that the planner automatically suggests the Globus supermarket. The last task is a reminder to catch the flight from Zweibrücken; it takes 31 min to go there from the office. Now compare the times and distances with the adaptive view for Munich on the right. The olives do not appear, since the store is too far away. For the milk, a different store in Munich is suggested and also the waterworld has been substituted by a local place. The adaptive reminder for the airport will happen earlier, since the estimated traveling time is now more than 4 h.

#### 6.2.2 The User Interface of VISTO

The VISTO user interface is designed to assist pedestrians in indoor environments and appears on wall-mounted public situated displays, which are spread across the building at decision points. The screen layout is split into three areas as shown in Fig. 18. On the top border, a status bar shows the name of the location and the registered activities which are used to search and filter the UBIDOO task tree for. The status bar also includes a personalized icon for each user that has been currently recognized by the Bluetooth scanner. The user icons are continuously updated after each scan (each scan takes approximately 15 s; single misses of a previously recognized device are ignored to stabilize the displayed list).

On the left-hand side the activities of the recognized users are shown as bluelabeled tabs. They show a list of tasks (actions), according to the hierarchic task tree structure in UBIDOO, which remind the user of his objectives depending on



Fig. 18 VISTO is designed to run on public displays and presents an adaptive list of goals (*left*) and animated walking directions (*right*)

the current spatial context. In the given example, one tab represents a shopping list of products, another tab a list of persons to visit. In the context of VISTO, each list item represents a navigational goal and provides some basic information to the user about the walking direction and distance (relative to the current display's orientation and location, which is known to the system through the YAMAMOTO environment model). The tasks are sorted by increasing distance, so that the closest goal appears on top of the list. If the display affords interaction by means of a touchscreen, keyboard, or mouse, the user can activate any task from the list by clicking on its arrow symbol. There can be only one active task per user. On the right-hand side, video frames present detailed walking directions to each active goal as a 3D animation. Each animation shows the route from the egocentric perspective of the user. The movie starts at the current location and the avatar is facing the respective display within the virtual environment model. The avatar follows the path until the destination or another display is reached. The movie loops until the user leaves the Bluetooth range of the display. If the user encounters a new display along the route, the playback of the next movie sequence is automatically triggered without any interaction. Finally, if the user has reached their destination, the current task will be deactivated. In summary, VISTO is a user-adaptive system that adapts its user interface according to the current activity of the user and proactively assists them in their wayfinding tasks.

### 7 Summary

In the chapter at hand, we have elaborated on the development of a map modeling toolkit for buildings in 3D as well as on a new indoor positioning system. The toolkit meets the demands of providing map-like material for pedestrian indoor navigation and allows for a quick and easy editing of indoor map material. In addition, it implements a route finding service for buildings. In parallel, we have implemented a new indoor positioning system. The positioning system is able to combine active RFID tags, infrared beacons, and GPS to calculate the users' position following the "always best positioned" paradigm in a resource-adaptive manner. On the basis of those core components, we introduced different visualization techniques for route descriptions using personal digital assistants or public displays available in the environment. The whole system, as explained in the chapter present, is able to seamlessly combine indoor and outdoor navigation tasks in such a way that the changeover between different localisation techniques is barely noticeable for the user. The results of the project led to several new external cooperations, e.g., Fraunhofer IESE Kaiserslautern and LMU Munich, which have been granted a research licenses for LORIOT and YAMAMOTO. Furthermore, they led to the foundation of a new spin-off company, schwartz&stahl indoor navigation solutions GbR (http://www.indoornavi.com) in order to bring the results of the project to market.

Within the project, we intensified our cooperation with the research group of the project AERA of the Collaborative Research Center 378, Resource-adaptive Cognitive Processes. During the last period, both projects, AERA and RENA, have been closely cooperating in the design of an experimental paradigm to investigate spatial navigation and wayfinding with the aim to optimize spatial instructions during the navigation task. The results of a series of experiments are summarized and reported by [28] in this volume and in detail in [1, 16, 17].

Acknowledgments This research has been supported by the German Science Foundation (DFG) in its Transfer Unit 53, Project RENA: Resource-Adaptive Navigation.

# References

- Aslan, I., Schwalm, M., Baus, J., Krüger, A., Schwartz, T. Acquisition of spatial knowledge in location aware mobile pedestrian navigation systems. In: Proceedings of the 8th international Conference on Human Computer Interaction with Mobile Devices and Services (Mobile HCI 2006) (pp. 105–108). ACM Press (2006).
- Bahl, P., Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. IEEE INFOCOM, 2:775–784 (2000).
- Baus, J. Ressourcenadaptierende Hybride Personennavigation. DISKI 268. Berlin: Akademische Verlagsgesellschaft Aka GmbH (2003).
- Baus, J., Krüger, A., Stahl, C. Resource-adaptive personal navigation. In: O. Stock, M. Zacanaro (Eds.), Mulitmodal Intelligent Information Presentation (pp. 71–93). New York: Springer (2005).

- 5. Beer, T. Ein System zur Gefahrenmeldung basierend auf Methoden des Wissensmanagements und des Semantic Web. Master's thesis, Universität des Saarlandes (2004).
- Brandherm, B., Schwartz, T. Geo referenced dynamic Bayesian Networks for user positioning on mobile systems. In T. Strang, C. Linnhoff-Popien (Eds.), Proceedings of the International Workshop on Location- and Context-Awareness (LoCA), LNCS 3479, Lecture Notes in Computer Science (vol. 3479/2005, pp. 223–234). Munich, Germany: Springer-Verlag Berlin Heidelberg (2005).
- 7. Department of Defense: Department of Defense World Geodetic System 1984. Tech. Rep. Third Edition, National Imagery and Mapping Agency (2000).
- Ehrl, J. Prototypische Entwicklung eines kontextsensitiven, regelbasierten Dienstes zur lokalen Gefahrenwarnung im automobilen Umfeld mit Technologien des Semantic Web. Master's thesis, Universität des Saarlandes (2005).
- Ekahau, Inc. Ekahau Positioning Engine. http://www.ekahau.com/products/ positioningengine/ (2004).
- Fickert, O. Ein Ubiquitärer Aufgabenplaner f
  ür Benutzergruppen. Tech. Rep., Unversit
  ät des Saarlandes (2005).
- 11. Fickert, O. Ein Ortsbezogener Termin- und Aufgabenplaner mit Routenwissen. Master's thesis, Unversität des Saarlandes (2007).
- Gholamsaghaee, E. Ein System zur Positionsbestimmung in Parkäusern mittels aktiven RFID-Tags. Master's thesis, Universität des Saarlandes (2007).
- Johanson, B., Fox, A. The event heap: A coordination infrastructure for interactive workspaces. In: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (pp. 83–93). Callicoon, New York (2002).
- Krüger, A., Butz, A., Müller, C., Wasinger, R., Steinberg, K., Dirschl, A. The connected user interface: Realizing a personal situated navigation service. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI 2004) (pp. 161–168). ACM Press, New York (2004).
- Müller, J. Beanspruchungsschätzung im Automobil mit Bayes'schen Netzen. Master's thesis, Unversität des Saarlandes (2005).
- Münzer, S., Stahl, C. Providing individual route instructions for indoor wayfinding in complex, multi-level buildings. In: GI-Days 2007, ifgi prints, Münster (2007).
- Münzer, S., Zimmer, H.D., Maximilian Schwalm Jörg Baus, I.A. Computer assisted navigation and the acquisition of route and survey knowledge. Journal of Environmental Psychology 26:300–308 (2007).
- Mußler, O. Verfahren zur Positionsbestimmung in Parkhäusern. Master's thesis, Universität des Saarlandes (2004).
- Schöttle, F. IPlay Blue: Interaktive Displays mit Bluetoothbasierter Benutzererkennung. Tech. Rep., Unversität des Saarlandes (2006).
- Schöttle, F. Automatisches Generieren von graphischen Wegbeschreibungen für Fußgänger. Master's thesis, Unversität des Saarlandes (2007).
- Schwartz, T., Brandherm, B., Heckmann, D. Calculation of the user-direction in an always best positioned mobile localization system. In: Proceedings of the International Workshop on Artificial Intelligence in Mobile Systems (AIMS) (2005).
- Stahl, C., Haupert, J. Simulating and evaluating public situated displays in virtual environment models. In T. Pederson, H. Pinto, M. Schmitz, C. Stahl, L. Terrenghi (Eds.), International Workshop on Modelling and Designing User Assistance in Intelligent Environments (MODIE 2006) (pp. 32–35) (2006).
- 23. Stahl, C., Heckmann, D., Schwartz, T., Fickert, O. Here and now: A user-adaptive and location-aware task planner. In S. Berkovsky, K. Cheverst, P. Dolog, D. Heckmann, T. Kuflik, J. Picault, P. Mylonas, J. Vassileva (Eds.), Proceedings of the International Workshop on Ubiquitous and Decentralized User Modeling (UbiDeUM'2007) (pp. 52–63) (2007).
- 24. Wahlster, W., Feld, M., Gebhard, P., Heckmann, D., Jung, R., Kruppa, M., Schmitz, M., Spassova, L., Wasinger, R.: The Shopping Experience of Tomorrow: Human-Centered and Resource-Adaptive. In: M. Crocker, J. Siekmann (Eds.) Resource Adaptive Cognitive Processes (this volume), Cognitive Technologies Series. Springer

- 25. Ward, A., Jones, A., Hopper, A. A new location technique for the active office. IEEE Personal Communications 4(5):42–47 (1997).
- Wasinger, R., Stahl, C., Krüger, A. M3I in a Pedestrian navigation & exploration system. In L. Chittaro (Ed.), Proceedings of the 5th International Symposium on Human-Computer Interaction with Mobile Devices and Services (Mobile HCI) (pp. 481–485). Udine, Italy: Springer (2003).
- 27. Wickens, C., Hollands, J. Engineering Psychology and Human Performance. Upper Saddle River, NJ: Prentice Hall (2000).
- Zimmer, H.D., Münzer, S., Baus, J.: From Resource-adaptive Navigation Assistance to Augmented Cognition. In: M. Crocker, J. Siekmann (Eds.) Resource Adaptive Cognitive Processes (this volume), Cognitive Technologies Series. Springer