The Connected User Interface: Realizing a Personal Situated Navigation Service

Antonio Krüger
FR Informatik
Saarland University
D-66123 Saarbrücken
krueger@cs.uni-sb.de

Andreas Butz
FR Informatik
Saarland University
D-66123 Saarbrücken
butz@cs.uni-sb.de

Christian Müller
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
cmueller@dfki.de

Christoph Stahl
FR Informatik
Saarland University
D-66123 Saarbrücken
stahl@cs.uni-sb.de

Rainer Wasinger
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
wasinger@dfki.de

Karl-Ernst Steinberg
BMW Research
Hanauer Strasse 46
D-München 80992
karl-ernst.steinberg@bmw.de

Andreas Dirschl
BMW Research
Hanauer Strasse 46
D-München 80992
andreas.dirschl@bmw.de

ABSTRACT

Navigation services can be found in different situations and contexts: while connected to the web through a desktop PC, in cars, and more recently on PDAs while on foot. These services are usually well designed for their specific purpose, but fail to work in other situations. In this paper we present an approach that connects a variety of specialized user interfaces to achieve a personal navigation service spanning different situations. We describe the concepts behind the **BPN** (BMW Personal Navigator), an entirely implemented system that combines a desktop event and route planner, a car navigation system, and a multi-modal, in- and outdoor pedestrian navigation system for a PDA. Rather than designing for one unified UI, we focus on connecting specialized UIs for desktop, in-car and on-foot use.

Categories and Subject Descriptors

H.5.2 [**HCI**]: User Interfaces—*User-centered design*; H.5.3 [**HCI**]: Group and Organization Interfaces—*Collaborative computing*

General Terms

Human Factors

Keywords

Pedestrian Navigation Systems, Ubiquitous Interfaces

1. INTRODUCTION

Navigation services are considered to be one major candidate for making 3G telecommunication technology a commercial success. While in-car navigation systems and desktop route finders are already popular and wide spread, PDA-based navigation services for pedestrians have only recently entered the marketplace. The special needs for pedestrian navigation systems have been widely acknowledged and tested in several research pedestrian navigation systems (e.g. [1, 7, 5]). This research has explicitly stressed the importance of landmarks, 3D-graphics, and multi-modal interaction. Although the development is promising, the commercially available systems are still lacking most of these features. One reason is that cartographic material is mainly being designed for car navigation systems, which is inadequate for pedestrian use. Furthermore, today's PDA-based navigation systems are not integrated with already available navigation services (e.g. web and car services).

We are convinced that designs aimed at one single device and user interface (e.g. the unified interface described in [10]) will not work for different navigational situations (e.g. at home, in the car, on-foot), because the complexity of each navigational situation usually asks for different user interface ergonomics. We describe in this paper an integrated navigation service that makes use of different hardware and software-platforms, to provide a seamless service for travellers in different situations. In particular, we describe how the different interfaces involved -on a desktop PC, a car navigation system and a PDA- have to be modified and coordinated by a navigation server to achieve this goal. In this paper our goal is to investigate the necessary representations, technical platforms and interaction modes. To better understand our motivation, let us have a look at the following scenario:

Mr. S from Saarbrücken plans a business trip to his new customer Mr. M in Munich. He starts to prepare the trip in his office while sitting in front of his desktop computer. First, he logs into the personal navigation web server, which provides him with information on earlier trips and travel itineraries. Instead of explicitly typing in Mr. M's street address, he can simply use Mr. M's electronic business card that is already stored on Mr. S's computer to specify the travel destination. The server verifies the address and provides Mr. S with a complete travel itinerary, including the trip from Saarbrücken to Munich with the car and suggestions where to park near Mr. M's office. Assuming that Mr. S will use the suggested park house, the system also returns outdoor path directions from the park house to Mr. M's company, and even indoor directions that lead Mr. S directly to Mr. M's office. After having received details on his route, Mr. S uses the server to book a hotel for the night and chooses from a list of suggestions for evening events that match his interests. This information on the travel itinerary is

IUI'04, January 13–16, 2004, Madeira, Funchal, Portugal.

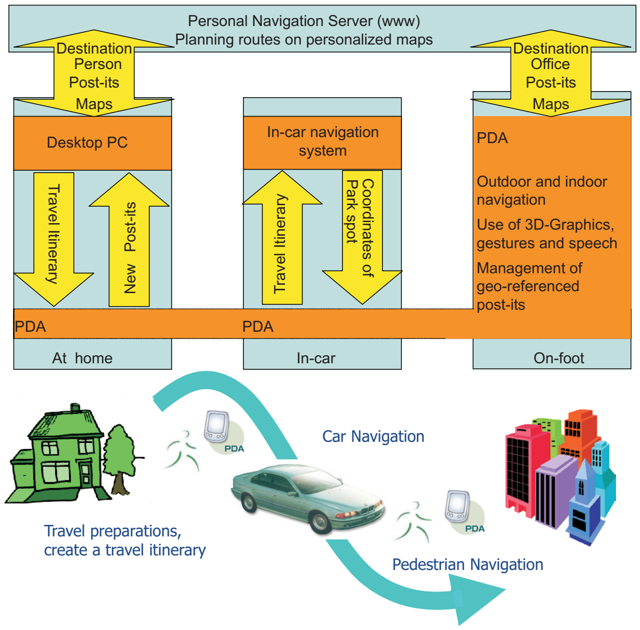

Figure 1: Different situations during a navigational task and the role of the PDA as the link between situations

then downloaded to Mr. S's PDA. When entering the car, the PDA is used to program the in-car navigation system at the push of a button, making the cumbersome input of address details unnecessary. The car navigation now takes control and guides Mr. S safely to his destination in Munich. After having parked at the designated location, Mr. S uses the PDA to navigate the remaining path to Mr. M's company. The PDA mainly uses speech and 3D-graphics to provide adequate information on landmarks and streets. Since Mr. S is one hour early, he decides to have a meal in a nearby restaurants. He uses speech and pointing gestures to ask the PDA for a recommendation, and the PDA points him to a good Italian restaurant around the corner. After the meal, Mr. M is able to record a short geo-referenced voice memo on the PDA that will later remind him of the good quality and location of the restaurant. Once at the building and after having registered at the security desk of Mr. M's company, Mr. S continues to use his PDA indoors and in the same way, until he reaches Mr. M's office. After a successful business meeting Mr. S uses the PDA to find his way back to the car. Finally, he heads off to his hotel and looks forward to a pleasant night in Munich.

In the following sections, we describe the project **BPN** which was jointly developed by BMW and the DFKI (German Research Center for AI). We first clarify what we mean by 'personal navigation' and distinguish between the different situations of a navigational task. We then describe the different **BPN** components, i.e. the role of the navigation server, the pedestrian map generation, the in-car navigation system, and the PDA. We then share some of our experiences and insights that were gained from the use of the system. The paper finishes with an overview on related work and an outlook on future activities.

2. SITUATED PERSONAL NAVIGATION

Traditional navigation systems are designed to work for a specific platform in a well defined environment. A typical example for such a class of systems are web-based route finders. These services are optimized for the PC and usually provide only little support for other devices (e.g. PDAs), relying instead on the directions being printed on paper. Another prominent class are car navigation systems, which can be divided into two subclasses. Whereas the subclass of *built-in* navigation systems, often shipped with the car, are restricted to supporting the user while driving, *PDA based* navigation systems (e.g. the tomtom navigator[1]) can also be used outside the car. The advantage of in-built systems is a higher positioning accuracy and very good usability under driving conditions due to a larger display and specialized input methods, e.g. the BMW I-Drive Controller. PDA based systems are more flexible, but rely on the same maps for in-car and on-foot conditions, causing suboptimal results when providing route descriptions for pedestrians.

As Figure 1 shows, we investigated three different types of *situations* in which navigational services may be of interest. We aim 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 travellers to prepare their trips at home in order to obtain route directions and maps, to choose personalized events of interest at their destination, to book an appropriate hotel, and to retrieve further information like the current weather. This information is collected and stored in a *travel itinerary* (see Figure 3 and section 3.1) for each trip. With the help of the PDA, travellers can also make use of their travel itinerary in the car, and as pedestrians on-foot, both inside and outside of buildings.

The PDA is the central component in our approach. However, rather than thinking of the PDA as a unified user interface that can be used to access information in different situations, the PDA is instead primarily used as a connecting link between different situations and interfaces[2]. Figure 1 exemplifies this idea: 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 of a direct internet connection. In the car the PDA connects locally to the car navigation system, which in turn allows us 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 traveller to the selected destination. Before leaving the car, the PDA receives the actual park spot coordinates, which are added to the travel itinerary and which may help to find back to the car later. On foot the PDA plays a much more vital role. It displays the 3D-map information included in the travel itinerary and guides the traveller with verbal and graphical route directions. It can also be used to store geo-referenced user data (e.g. voice memos) and to respond to multi-modal requests regarding landmarks in the environment (e.g. "What is *this* building"). If required, the PDA can connect to the personal navigation server and receive updated information (e.g. on path directions), which is important when the user gets lost.

In the following sections, we will describe in more detail the components of the **BPN** that are involved in providing for situated personal navigation.

3. THE BMW PERSONAL NAVIGATOR

3.1 Intelligent Travel Itinerary Management

A BPN website has been realized to provide an interface for planning trips through the internet. It incorporates several exter-

[1] http://www.tomtom.com

[2] Assuming that the PDA always resides with the traveller.

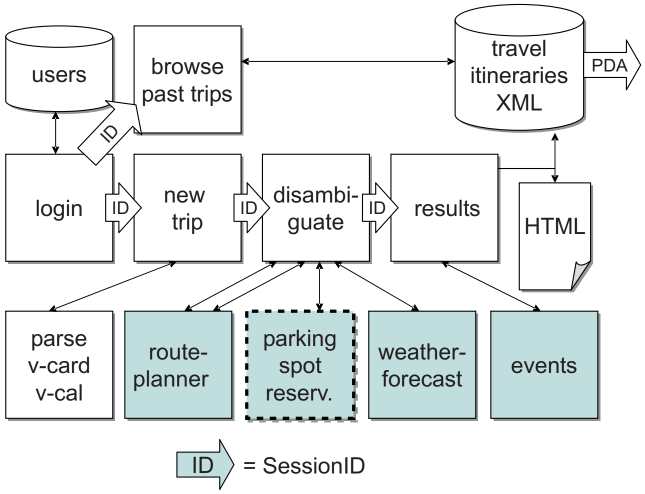

Figure 2: Overview of the trip management modules

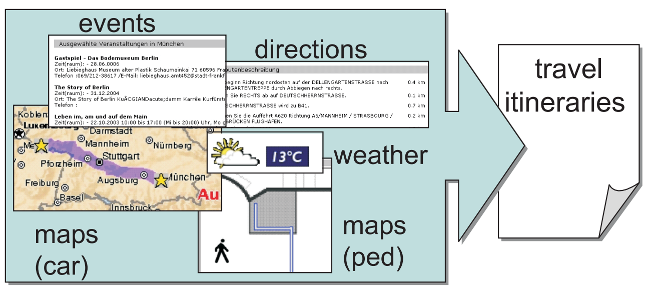

Figure 3: All information relating to the trip is stored in the travel itinerary

nal web-services including a route planner, a weather forecast service, and an event-service for identifying activities that the user may wish to undertake in their spare time.

When registering, a new traveller provides his/her name and address, as well as an interests profile based on a list of event categories (e.g. sports, concerts, or exhibitions). The route planning process is designed to be intuitive and to require minimal user input. When planning a new trip, the standard start address (e.g. the traveller's office or home address) can be selected by clicking a single button. A second alternative is to select the start address from a list of previous trips. If none of these are suitable, the address can also be typed in manually. For providing the target address, an additional alternative is available: the user can select a contact from their desktop organizer and upload it to the server. The format of the contact file is *v-card*, a standard *ASCII*-based contact-export-format that is provided by most of the common organizers (e.g. MS-Outlook and Palm Desktop). The server parses the v-card file and automatically fills in the forms on the web interface. The relevant information encompasses the street address for car route-planning (and outdoor pedestrian route-planning) as well as the floor and the room number for indoor route-planning. Providing the date and the time of the trip works similarly: the user uploads a *v-cal* file representing the calendar entry for this trip. Of course, both the target address and the date and time can be typed in manually as well. Next, the user specifies whether the trip should be planned up to the given target address or up to a parking area nearby. In the latter case, the address of the parking area and the original target address serve as input for the pedestrian route planner. It generates the route from the park spot to the target building and the directions inside the building (see section 3.2). The address of the parking area is currently preselected, but this is intended to be replaced by a park-spot-reservation-service in future extensions of the system. After providing all the necessary information in a way that best suites the user, the route-planning process is activated. The server calls a standard route-planning-service[3] for the maps and driving directions, a weather-forecast-service [4] for the weather in the target city, and an event-service[5]. The results (in HTML format) are parsed and relevant information (e.g. directions and graphics) are extracted. The system's own HTML templates are then filled with this information on one single page. Moreover, the results of the event-service are filtered according to the interests profile of the user.

Figure 2 summarizes this procedure. The traveller logs in and creates a new trip. S/he uploads a v-card and a v-cal file to the server, which parses the files and fills out the forms accordingly. The server then calls the external services (colored boxes). Note that there may be several iterations due to ambiguous results. For Munich as a target city for example, the external route-planner returns several entries because there are smaller towns in Germany with the same name. If the provided street name does not help identify the correct city (and there is no postal code provided in this instance of the v-card file), then the list of alternatives is presented to the user for selection.

Once the start and target address are disambiguated, the extracted maps and travel directions are unified with the weather information. The event service is then called with the target city as an input parameter. The results are again parsed and filtered according to the date of the trip and the user's interests profile. Currently the system uses nine main interest categories (excursions, exhibitions, stage, fun, children, concerts, party, sports, and top events). Each of these is divided into subcategories (sports is for example split into the categories football, motor sports, tennis, water sports, winter sports, and miscellaneous). The boxes in Figure 2 represent the modules of the server component, which are implemented as *cgi-scripts* using *perl*. The intermediate data is communicated among the modules using the single cgi-parameter *session-id*. The rest of the data is stored on the server where the modules then have access to this information using the session-id key. In this manner, the number of cgi-parameters is kept small although more and more information is collected.

The service *parking spot reservation* is depicted with dotted lines because it has not been implemented yet. On the results page, all of the information is incorporated and presented consistently. On the basis of these travel itineraries, the user can prepare for the trip, pack the right clothes, and schedule some free time undertakings. The travel itineraries are stored on the server for later use. They consist of a car section with the above described information about driving, weather, and events, and a pedestrian section with both outside and inside maps of the target building (see Figure 3). These are all represented in one single XML file to facilitate easy synchronization with the PDA.

Pedestrian navigation together with car navigation serves as an example for combining various means of transportation within a single framework. The approach described here allows for generalization by adding for example flights or public transportation.

[3]http://www.mapquest.de
[4]http://www.mc-wetter.de
[5]http://www.vivimed.freizeit.de

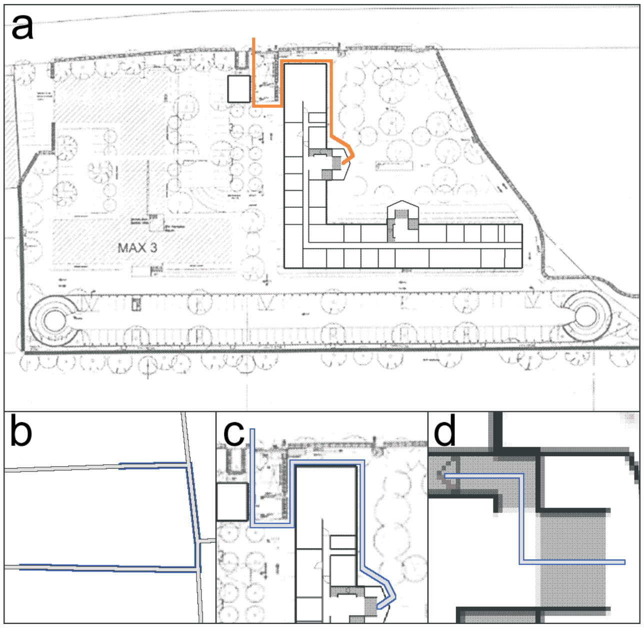

Figure 4: Maps for the PDA (a) have a high resolution and show details of the surroundings. Maps for the web preview (c) use a different map style and a tighter cropping. Bottom row: b) on the street, c) close to a building, and d) inside a building

3.2 Route Finding and Map Generation for Pedestrians

The route finding and map drawing processes for the pedestrian maps in **BPN** are based on commercially available street data (Navtech Navstreets Street Data Premium Germany, ArcView Format) and an open source Geographical Information System (GRASS version 5.0[6]). Route data within specific buildings is modelled manually in GRASS.

The street data contains a network of streets or street segments. For each of these segments, a number of properties including street name and classification are stored. This helps to determine whether a specific segment can be used by pedestrians, and what the segments postal address range is. Although using a commercial data set involves dealing with very large amounts of data, this is the only way to provide a map generation process that will scale beyond a few hand-modelled localities. The data we used is available for most parts of the world.

In order to perform calculations and visualizations with the street data, we decided to use the open source GIS GRASS, which was freely available and could be modified and extended to suit our needs. GRASS runs on several Unix platforms and consists of many small components and shell scripts, which can be combined to automate complex processes. It provides import routines from many GIS data formats, particularly the ArcView format that we use. We extended GRASS by incorporating a route planning module, by importing and reducing the street data, and by defining special rendering styles for the map drawing modules of GRASS, in order to generate different types of pedestrian maps.

For route planning within buildings, the path networks for all of the floors of a building have to be acquired. This is done by scanning or converting architectural plans of the building and importing them as bitmap layers into GRASS. With the map editor of GRASS, the path network can then simply be drawn on these bitmap backgrounds. A substantial advantage of this procedure is that instead of being restricted to correct architectural plans, all kinds of existing map material can be used. Maps drawn by designers might be substantially distorted or out of scale, but might represent the building much better from a perceptive point of view. Each floor of a building is represented in a separate GIS layer. Stairways between floors are split into two parts, each of which belong to the floor from which it is accessed. Connection points between different maps (either between different floors or between the ground floor and the street data) are kept in a separate database and are used for route planning across several maps.

The creation of path networks within buildings is a manual process, and for obvious security reasons there is no central database of buildings. It will be the building's owner's responsibility to provide map information if they want to participate in a service such as **BPN**, and if they decide to do so, the maps can be derived in a standardized way from existing architectural plans. In a sense, this prevents arbitrary scalability of the system, but there is no other way of obtaining this kind of detailed data. One possible market group for a product such as **BPN** would include large companies with extensive office buildings. These companies almost always employ a public relations department. In many cases, these people will already have produced map material of the site for brochures or overview plans in the building. For them, it is just a matter of exporting these graphics in a different format and specifying the possible routes within the maps in an additional step. Thus, by choosing the right target group and entry point, data for **BPN** comes at almost no additional cost.

Within the street networks and the path networks of buildings, routes are searched using a variation of Djikstra's algorithm. The weights of the street segments are determined by their length, but also by the number of turns a street makes, and by the number of opportunities to take a wrong turnoff. This favors the creation of routes with a lower risk of getting lost. The result of the route planning module is a new GIS layer containing the planned route for map drawing, as well as an XML description of the route for other components of the **BPN**. Route planning across several maps is facilitated by the database of connecting points and results in a route containing several subroutes, each with its own map.

The route layer generated in the planning process is used for generating the final 2D pedestrian map. The map drawing module of GRASS permits the consecutive drawing of several layers. The background is provided by the whole reduced street network for outdoor maps, and by the bitmap plans of building floors for indoor maps. On top of this background, the actual route is drawn as a framed line. The width and color of the line segments and their frames are defined in drawing style definitions. The map region is determined automatically from the generated route through a parameter describing how much of the surroundings should be provided. We can therefore generate a small preview map for the web interface from the same routing data, showing a tight cropping of the generated route, but also a higher resolution map for the 3D visualization, providing a larger context of the surroundings (see examples in Figure 4).

One of our goals was to provide pedestrians with 3D-graphics of the landmarks. For the time being there is no data-set commercially available that contains this information. For this reason we had to build up suitable 3D-representations of the landmarks by hand. Together with the respective maps, these are stored in the above mentioned XML file and added to the travel itinerary.

[6] http://grass.itc.it/

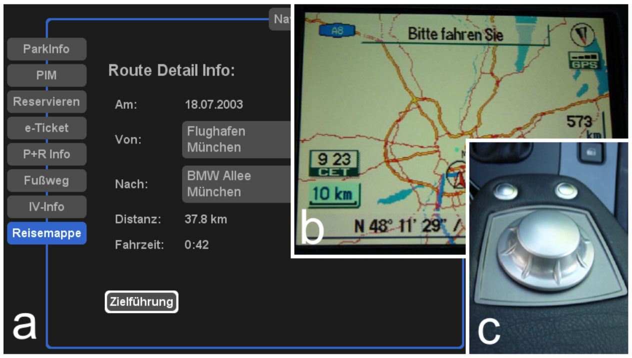

Figure 5: Screen shots of the BPN in-car user interface. An overview provides information on the actual travel itinerary (a). Maps are used to give directions (b). The main input device in the car is the I-Drive controller (c)

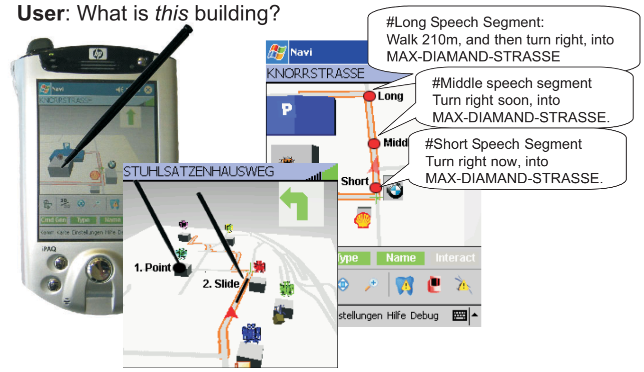

Figure 6: User input through speech-gesture combined interaction, and presentation output through adapted speech utterances

3.3 In-Car User Interface

A goal in designing our interfaces for different contexts (e.g. at home, in-car, on-foot), and different hardware requirements (e.g. screen sizes, performance), was to keep the individual interfaces as similar in design as possible. Since most of the in-car user interface already existed, design centered on the desktop and PDA interfaces. Our main goal was to keep the time required for program learnability to a minimum, while still allowing for the incorporation of each of the individual devices's hardware capabilities.

The in-car navigation system used in **BPN**, comprises software from Alpine Electronics, an on-board server running Apache Tomcat, and a Bluetooth access point. One aim of the in-car navigation system is that of providing the user with navigational aid while driving to a given destination. Navigational aid is provided in the form of both visual and acoustic output. The primary visual component is that of the navigation map, which contains a highlighted route for the active trip. The acoustic output is in the form of speech, and includes distance, direction, and street name information.

Upon entering the car, a user will typically start the car (which also starts the on-board navigation server), and then upload the travel itinerary from the PDA to the car, via the on-board bluetooth access point. The required connection details are stored inside a parameter configurations file on the PDA, and include the car's IP address, the port number, and the local address of the required servlet application.

The content of the travel itinerary is presented to the user on the car's display (Figure 5a). Being larger in size than the PDA display it makes browsing the information of the travel itinerary much more comfortable. The on-board I-Drive controller may be used to select a trip (Figure 5c), which then generates and loads the appropriate map (Figure 5b). Now, the PDA can be turned off until the car reaches its destination. Then, the PDA synchronizes the current car coordinates with the car navigation system. This is particularly important, because there may be delays when starting the PDA's GPS device, or the GPS signal may not even be available, as in the case of the car having been parked deep inside a park house. A future feature of this system will see the PDA in use alongside the car server, while navigating to a destination. This will allow a "co-driver" to record additional points of interest such as cheap petrol stations, based on time- and geographical-stamps. This feature is currently only available when on foot.

3.4 Mobile User Interface

The mobile user interface exhibits a rich degree of multi-modal user interaction in the form of both user input, and presentation output. User input can take the form of speech or speech-gesture combined requests (as shown in Figure 6), in which the gesture input can be either 'intra' in that the user points to an object on the display through the use of a stylus, or 'extra' in that the PDA is used to point to an object in the real world. Presentation output is expressed in the form of route descriptions that are presented via 2D/3D graphic visualizations and audio segments.

3.4.1 Speech Synthesis and Recognition

The formant-based speech synthesizer[7] receives navigational data such as street names, distance, turning angle and landmark information from the XML files (stored in the travel itinerary), from which it then generates appropriate route descriptions. Long, middle and short phrases are created for each segment and presented to the user in combination with graphics at different locations along a segment (i.e. start, middle, and end). The user also has the ability to have the current route description repeated, with updated context-sensitive information such as remaining segment distance and surrounding landmarks. Figure 6 illustrates a typical output presentation incorporating both speech and graphics.

The speaker-independent recognizer incorporates both a set of static (pre-compiled) grammars, and a set of dynamically created grammars. The static grammars cover map control functionality (e.g. "zoom in"), trip queries (e.g. "where is my start?"), and simple multi-modal interaction with objects (e.g. "what is *that* [gesture]?"). To cover more advanced user interaction with street and landmark objects, we also have a set of three grammars which are dynamically created each time a map is loaded. The first grammar allows for interaction with landmark types (e.g. "what is the name of *this* [gesture] church?"), while the second allows for interaction with specific street and landmark names (e.g. "take me to *Poststrasse*"). A third grammar containing information analogous to that found in tourist pamphlets like 'description', 'opening hours', and 'cost', will allow for detailed interaction with the individual landmarks (e.g. "what are the *opening hours?*"). Our set of static grammars contain around 70 unique words (not all active at the same time), while the set of dynamic grammars currently contain an average of 20 to 30 additional words, depending on the number of landmarks present in the map. Words not present in our pool,

[7] IBM Embedded ViaVoice was used for synthesis and recognition.

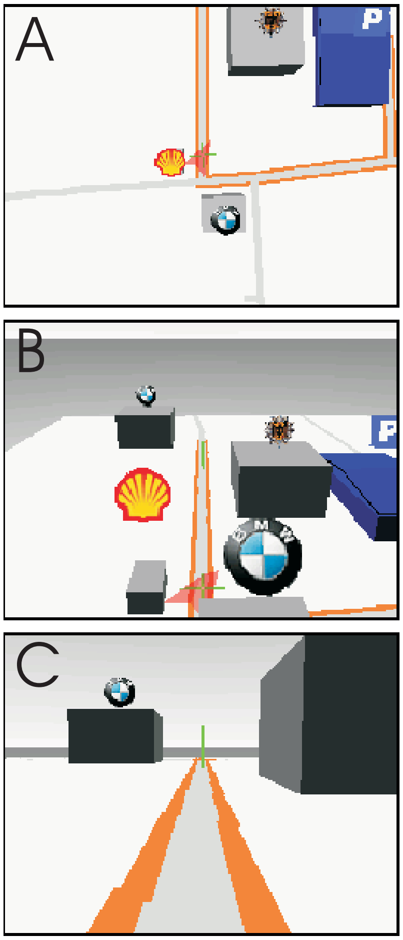

Figure 7: Examples of different perspectives available for pedestrian navigation in the BPN

have their baseforms automatically generated by the recognizer.

The speech grammars work in two modes. The first mode allows the traveller to manually specify which grammars s/he wants activated by pressing buttons on the display. The second mode called "active smart", records the user's request, and funnels this through to each of the grammars individually, searching for the best possible match. If no match is found, the confidence-value boundary is removed, and the result is chosen from a combination of the results returned from the first pass, and the single result returned from the second pass. Whereas the response time for the manual mode is negligible (under two seconds), the active smart mode currently has a worst-case response time of around 20 seconds.

The above strategy, in combination with our relatively small map sizes, and acoustic models designed for hand-held devices allow for robust rates of user-recognition.

3.4.2 3D-Graphics Generation

Providing pedestrians with 3D maps of their environment, especially when aligned to the user's 'heading' direction (in comparison to 'true north'), is known to help provide pedestrians with a good awareness of important landmarks [7]. Further advantages of 3D maps include the ability to roughly estimate the height of landmarks and the ability to see more of the path segment than with a simple top-down view. In order to maintain the highest level of flexibility, we decided to provide the mobile user with the possibility of switching between different perspectives. The traveller can choose between a top-down view of the scene (Figure 7a), an isometric view (Figure 7b), and an egocentric view (Figure 7c). It is also possible to change the orientation of the view, aligned to either the user's heading direction, or towards true north. With these requirements in mind, we evaluated several 3D engines and finally incorporated the Cortona embedded VRML browser[8] to build a PDA framework for the display of graphical route instructions for pedestrians.

The 3D-graphics consist of three layers. The bottom layer is textured with the map that has been generated according to the descriptions in the section 3.2. A second layer then places 3D landmark representations on top of the map, and the third layer consists of additional meta-graphical elements (e.g. arrows, markers for the user's position and flags for intermediate targets) that help to give appropriate route information at the right moment in time. The positioning mechanism has been decoupled from the visualization component, to allow for the use of several different localization techniques. Following [1], **BPN** uses GPS for outdoor use and infrared beacons for indoor use. It is for this reason that the maps we use can be represented in either geo- or pixel coordinates. The latter representation is especially useful to directly define the positions of infrared beacons on a map.

Transitions between different maps are automatically triggered when the traveller enters a certain region (e.g. if close to the entrance of a building). These transition points are also useful in informing the user of additional information, such as to register with a security officer at the entrance of a building.

A map-matching algorithm ensures that even with a weak GPS signal the most probable position is displayed on the map. The actual streetname is located at the top of the user interface, and several icons provide additional information (e.g. on the GPS signal strength). A special feature of the mobile **BPN** UI is the ability to record geo-referenced voice memos, notes and pictures. These are represented as 3D-icons along the road and can be reviewed at any time. The 3D-engine is also responsible for processing the stylus-gesture events generated by tapping the stylus on the PDA's display, which is described in more detail in the next section.

3.4.3 Gesture Interaction and Media Fusion

The mobile user interface accommodates for both intra and extra gestures. Intra-gestures are performed using the PDA's stylus, while extra-gestures require sensor information, such as the user's direction and speed, which can be obtained from the PDA's compass and GPS units respectively. With this available information, the user can point to objects in the distance and have the system understand what is being referred to. Two types of intra-gestures are currently recognized by our system. The first is a *point* gesture used to query landmarks. The second is a *slide* gesture used to query street names. These are shown in Figure 6. When the user touches a landmark, an object reference is returned by the VRML component, otherwise map-coordinates (longitude, latitude) are returned. Important to note, is that all gestures are mapped from the 2D screen to the 3D scene space.

A media fusion component is responsible for the fusion of speech and stylus gesture events, in which individual events are written to a blackboard and later unified to form a single modality-free result (for more details see [14]).

[8]http://www.parallelgraphics.com/cortona

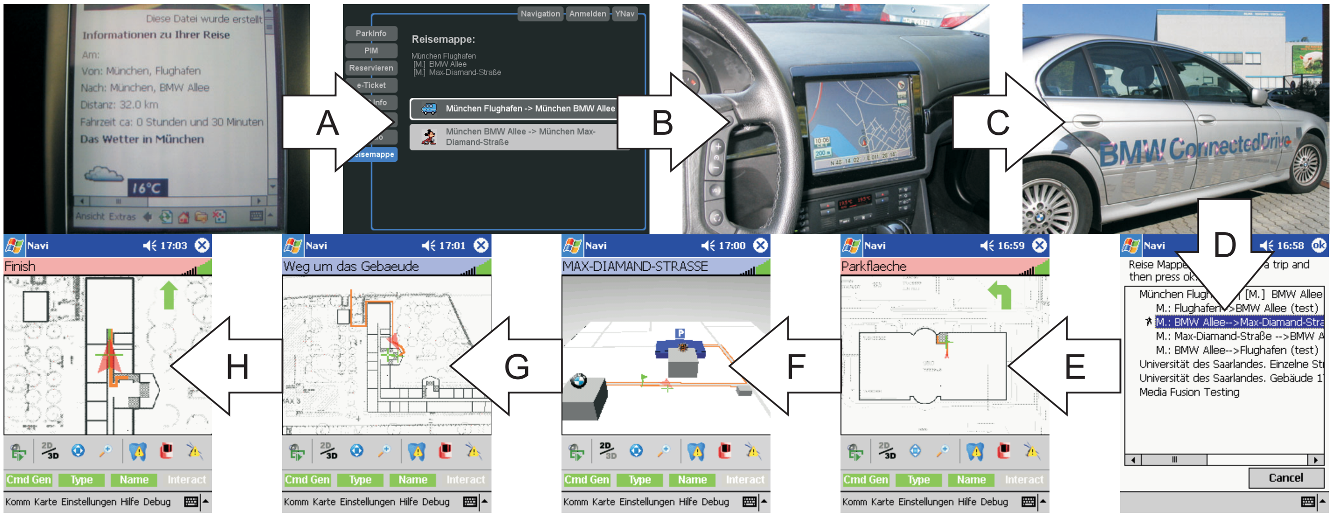

Figure 8: A test run of the system covering a route from Munich airport, to the office of one of the authors at the BMW research center

4. TESTING THE BPN

In order to evaluate the design and the functionality of the developed BPN prototype, we have run several test trials in Munich and Saarbrücken. These trials have provided us with valuable information that was used to redesign parts of **BPN**'s functionality. Figure 8 depicts the different stages of a test run from Munich airport to the office of one of the authors, at the BMW Research Center. In this case the traveller has already prepared the trip at home and synchronized the PDA with the corresponding travel itinerary. When entering the car, the PDA connects to the car server and transfers the travel itinerary to the car navigation system (Figure 8, step A). This currently requires a few button clicks on the PDA. We also considered realizing an immediate data transfer out of the travellers pocket. This is however not possible with todays technology, as the PDA has to be manually turned on before it can establish a Bluetooth connection with the car server. After the car has received the trip information, the in-car user interface takes control. The traveller browses through the content of the travel itinerary and selects a target destination ("BMW Allee", step B) and the car navigation system guides the traveller to the BMW main park house (step C). The car navigation system then prompts the user to continue their trip using the PDA again. The car coordinates are transferred to the PDA and the traveller selects the route entry from the travel itinerary that corresponds to the pedestrian part of the trip (step D). This starts the pedestrian navigation mode (step E), which guides the traveller from inside the park house to the street (step F). The required indoor localization for this was realized through the use of infrared beacons[9], installed at crucial intersections along the demo route. When on the street, user-positioning is achieved through the use of GPS. We decided to use a Bluetooth GPS receiver to minimize cabling and bulky PDA jackets. Another advantage of this choice is that the device comes with its own battery (in contrast to a CF-card GPS), allowing us to run the receiver separately from the PDA. This helps to avoid cold start problems that arise when a GPS receiver is turned on after being off for a longer period of time.

As described in section 3.2, different maps for indoor and outdoor navigation are used. When leaving a building (here the park house), the PDA switches to an outdoor 3D-view, including 3D-representations of important landmarks. This facilitates multi-modal user requests and easier identification of environment objects. While testing the system, we noticed that when a path is close to a building (see step G), it is more convenient to switch to a top-down view to avoid occlusions. In step F, the traveller enters the BMW Research building and the localisation method switches from GPS to infrared again. The zoom functionality of the **BPN** proved very useful in distinguishing details on the map, especially when inside buildings (see the last image of the sequence in Figure 8).

The necessary route data for the navigation consists of an XML file, the map graphics and 3D models. The size of the uncompressed XML file in our test-run scenario was around 22 KB. It contains the description of five building landmarks and 26 beacon coordinates. The 3D models of the buildings are stored in separate VRML files, each around 5 KB in size and consisting of approximately 50 triangles, with additional bitmaps for the icons requiring 14 KB each. The eight maps (1024*1024 pixels each) consume 326 KB (in GIF format). To start the navigation on the PDA (HP iPAQ 5400), it takes about 10 seconds to read the XML file and load all 3D models and image textures. Once the scene is loaded, it is rendered in 3D at an average rate of 5 frames per second. The response time is acceptable since the user position is updated by GPS and IR once every second.

To assess the limitations of the system in a stress test, we modelled the Saarland University campus in more detail. The resulting route file has an uncompressed size of 10 KB, and contains 21 buildings and landmarks. The VRML files for the landmarks and buildings contain around 700 triangles and have an uncompressed size of 85 KB. The total compressed size of the files is 31 KB. Loading this scene takes around 60 seconds, and the frame rate is between 2 and 4 fps. Whereas the overall response time of the system after loading is still reasonable, the simultaneous presentation of meta-graphical elements and speech introduce increased frame dropouts. By further increasing the number of landmarks and adding more complex speech grammars, the system eventually runs out of memory. These problems could be solved by using a more recent and more powerful PDA.

[9] For this trial 26 Eyeled-beacons were used.

5. RELATED WORK

There has been a lot of work on tourist and navigational guides (e.g. [2, 11, 8]) in recent years. So as not to loose this paper's focus, we will in this section only describe related approaches that either a) combine several situations and user interfaces or b) use similar advanced technologies on mobile devices (i.e. 3D-graphics and speech technology) to provide route directions. **REAL**[1] is a hybrid navigation system that provides navigation for indoor and outdoor situations. The focus of **REAL** was on the adaptive generation of graphics in response to technical and cognitive resource limitations. **REAL** combined a large screen information kiosk with a small PDA during the navigation task. In the **Smartkom** framework [13], a mobile assistant (based on a PDA) was developed to process multi-modal input and output (speech and gesture) during a navigational task. The presented information combined maps, natural language and a life-like character. **Smartkom** can be used in the car, but was not designed for indoor use. In **Deep Map** [9] a tourist guide for the city of Heidelberg was developed. Similar to **BPN** it was possible for the tourists to plan their trips beforehand, but in contrast to **BPN**, which uses standard map material, non-standard maps and 3D-models (derived from arial scans) were used. Deep Map is also only targeted at pedestrians and is not intended for in-car use.

The second class of related systems mainly use 3D graphics and/or natural language processing on the mobile device, to provide users with information on their environment. **TelMaris** [6] is similar to BPN in combining 3D-models with 2D maps to provide boat tourists with information on the city of Tønsberg in Norway, to locate appropriate hotels. The focus of **TelMaris** lies on the graphics component, and does not cover natural speech interaction. Therefore, neither indoor usage nor natural speech interaction is possible with **TelMaris**. The first system that combined natural language in- and output together with 3D-graphics on a PDA was the **Mobile Reality** system[12]. It provides indoor information on persons and on equipment of a building. The 3D-component helps to localize offices, while natural speech was used to address queries. **Mobile Reality** was designed for indoor use only.

In comparison to systems like **GALAXY**[4] and **Quickset**[3] which process multimodal HCI interaction through a client-server setup, BPN processes all user interactions locally on the PDA, effectively allowing a user to remain offline for the entire trip (assuming they do not lose their way).

Although **BPN** was inspired by most of the above described systems, and therefore shares several features with them, it is unique in the way that it combines three different contexts, i.e. at home, in the car, and on-foot, in a single situated navigation service.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the prototype of a navigation service that spans three different contexts: at home, in-car and on-foot. The service combines a desktop event and route planner, a car navigation system, and an in- and outdoor pedestrian navigation system for PDAs. Instead of using a unified user interface, we focused on the synergetic effect that arises when connecting the specialized UIs together. We presented in detail the required server-, car- and PDA- components. Although **BPN** is entirely implemented, we are continuing to develop the system in two directions: a) by extending the server concept of the travel itinerary towards personalizing maps that aggregate all personal information collected during a trip into one single representation, and b) by enhancing the mobile component to include touristic information that can be dynamically downloaded on request. Furthermore, we intend to generalize the concepts behind the **BPN** framework, to allow for the access of personalized information in instrumented environments.

7. REFERENCES

[1] J. Baus, A. Krüger, and W. Wahlster. A resource-adaptive mobile navigation system. In Y. Gil and D. B. Leake, editors, *IUI 02 – 2002 International Conference on Intelligent User Interfaces*, pages 15–22, San Francisco, California, 2002. ACM Press.

[2] K. Cheverst, N. Davies, K. Mitchell, and P. Smith. Providing tailored (context-aware) information to city visitors. In *AH 2000*, pages 73–85. Springer, 2000.

[3] P. Cohen, L. Chen, J. Clow, M. Johnston, D. McGee, J. Pittman, and I. Smith. Quickset: A multimodal interface for distributed interactive simulation. In *Proc. of UIST 96*. ACM, 1996.

[4] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, and S. S. andb V. Zue. Galaxy: A human-language interface to on-line travel information. In *Proc. ICSLP '94*, pages 707–710. ICSLP, 1994.

[5] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring mars: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers and Graphics*, 23(6):779–785, December 1999.

[6] C. Kray, K. Laakso, C. Elting, and V. Coors. Presenting route instructions on mobile devices. In *Proceedings of IUI 03*, pages 117–124, Miami Beach, FL, 2003. ACM Press.

[7] K. Laakso. Evaluating the use of navigable three-dimensional maps in mobile devices. Master's thesis, Helsinki University of Technology, Helsinki, November 2002.

[8] A. C. Long, S. Narayanaswamy, A. Burstein, R. Han, K. Lutz, B. Richards, S. Sheng, R. W. Brodersen, and J. Rabaey. A prototype user interface for a mobile multimedia terminal. In *Conference companion on Human factors in computing systems*, pages 81–82, New York, 1995. SIGCHI, ACM Press.

[9] R. Malaka and A. Zipf. Deep Map - challenging IT research in the framework of a tourist information system. In *Information and communication technologies in tourism 2000*, pages 15–27. Springer, 2000.

[10] T. Nishimura, H. Itoh, Y. Yamamoto, and H. Nakashima. A compact battery-lesss information terminal (CoBIT) for location-based support systems. In *Procceedings of SPIE 2002*, pages 80–86, 2002.

[11] E. Not, D. Petrelli, M. Sarini, O. Stock, C. Strapparava, and M. Zancanaro. Hypernavigation in the physical space: Adapting presentations to the user and to the situational context. *The New Review of Hypermedia and Multimedia*, 4:33–45, 1998.

[12] H. W. Stuart Goose, Georg Schneider. Mobile reality: A pda-based multimodal framework for synchronizing a hybrid tracking solution with 3d-graphics and location-sensitve speech interaction. In *Proceedings of Ubiquitous Computing 2002*. Springer, 2002.

[13] W. Wahlster, N. Reithinger, and A. Blocher. SmartKom: Multimodal communication with a life-like character. In *Proceedings of EuroSpeech 2001*, volume 3, pages 1547 – 1550, Aalborg, Danmark, 2001.

[14] R. Wasinger, C. Stahl, and A. Krüger. Robust speech interaction in a mobile environment through the use of multiple and different media input types. In *Proc. of EuroSpeech 03*, pages 1049–1052. ISCA, 2003.