Here and Now:
A User-Adaptive and Location-Aware Task Planner

Christoph Stahl, Dominik Heckmann, Tim Schwartz, Oliver Fickert

Saarland University
66123 Saarbrücken, Germany
[stahl, dominik, schwartz]@cs.uni-sb.de

Abstract. In this paper, we present a ubiquitous location-based task planner that integrates a to-do list and a schedule/calendar with route knowledge and adapts both its view and alarms to the user's current situation. The task planner is hosted on a web server and can be accessed from everywhere via a mobile Web terminal, such as a mobile phone or notebook, or a public display infrastructure which recognizes the user by their Bluetooth-device. Tasks can be localized by specifying a certain location where the task can be accomplished, such as an office or a store. Alternatively, a category can be chosen from an ontology that includes activities like shopping, sports or traveling by airplane. Since the task planner is likely to include dozens of tasks for the near future, it is too large to be browsed on the go. Therefore the planner implements a 'here-and-now' view, which adapts to the current time/date and location of the user. Based on knowledge about the purpose, address and opening hours of locations and routes, the task planner is able to filter for tasks that can be accomplished nearby, considering the time to reach the location and other deadlines. A second feature is an adaptive reminder, which considers the time that is needed to travel to the specified location of a task.

1 Introduction

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 to 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 the designers and the users, it has obvious drawbacks that we want to address in this paper.

Consider the following scenario, where a user C who is working in Saarbrücken creates a new entry in his calendar that reminds him to take a flight that starts in two weeks from Zweibrücken at 6:35 pm. C has to consider the time that is required to drive to the airport, lets say 30 minutes, and to prepare himself for the trip, another 30 minutes. The check-in procedure requires that C arrives at the airport one hour earlier, so he will set his alarm to 4:05 pm. However, two weeks later C is on a business trip in Munich, and the low-cost flight could not be changed. Now he is more than 4 hours from Zweibrücken. The alert can't recognize the new situation and would remind C much too late. In other situations, when the user is already on site, the problem is reversed and the alert will be activated too early. This is not too much of a problem since the user can regularly check their calendar for upcoming events. However, on mobile devices the display size is a limiting factor and the events have to be checked day-by-day in a cumbersome procedure. Also, the to-do list has to be additionally browsed for unfinished tasks. Again, the display is usually limited to a few tasks, so the user always has to navigate step by step through the complete list of items.

In the scenario, C also requires driving instructions from Munich to Zweibrücken. He has to look up the street address of the airport from the website and to manually enter it into a route planner or onboard navigation system. In case of an unfamiliar rental car, C might even need to consult the manual of the car in order to do so.

Further shortcomings arise from the P in the PIM concept: the tools are strictly limited to personal use. Let's assume another scenario: As C's girlfriend D recognizes that there is no milk in the kitchen, she adds it to her mobile phone's to-do list. C, who is in a meeting with a customer close to a shopping center, could quickly buy it on his way home - if he would know about both the milk and the store nearby.

Our goal is to ease the use of reminder tools by reducing the users' cognitive load and to simplify the workflow between PIM and navigation systems through the principles of user-adaption and location-awareness. We adopt a Web 2.0-based approach similar to Google's calendar application that offers ubiquitous access via mobile internet terminals and allows for collaboration between multiple users through the sharing of tasks and events. The necessary ubiquitous user- and location modelling infrastructure is contributed by the GUMO and UbisWorld ontology. Furthermore, we utilize a commercial Web service for route planning.

In the following sections, we will introduce relevant concepts and related work, before we present our own task planner and describe the modelling of tasks, users and locations. We explain how the reminder function and the task view adapt to the current location of the user and close the paper with an outlook and conclusions.

2 Basic Concepts and Related Work

In order to ensure that the whole variety of the distributed systems use the same semantics for the communicated concepts we base this work on ontologies and especially on semantic web technologies. The initial idea behind the semantic web is to annotate documents in the World Wide Web with semantic information by the use of ontologies, see [Fensel et al., 2003]. Languages like RDF and OWL have been designed to allow the author to declare the page contents as resources. Statements can be included in order to describe the semantics and relationships between resources using ontologies. By doing so, the web may be used in the sense of a very large distributed knowledge base. The goal in mind is the development of a machine-processable and human readable language, see [Studer et al., 2003], for representing and exchanging knowledge about a specialized domain. For example in [Stahl and Heckmann, 2004] we show how to use the semantic web technology for ubiquitous hybrid location modeling.

The concept of web services provides an interesting solution to solve the integration problem among heterogeneous application systems. It can be seen as a kind of standardized software technology to integrate and share various systems. It has the advantage of flexibility by perfectly defining standard specifications for mutually sharable data among distributed systems. So the web services provide the advantage that they can transparently access any web servers in any place with any device and at any time. The web services architecture combines three essential roles: service provider, service registry and service requestor.

During the course of our work, a reactive calendar on a mobile phone has been implemented by Glaubitt [Glaubitt 2006] that overcomes the limitations of a static calendar in a mobile environment through three types of situation-dependent events. First, the system dynamically adapts the reminder for a task according to the distance between the user and the location where the event is going to take place. Secondly, events can be triggered as the user enters a certain place. The third type of event is defined by its participants. For example, a meeting event begins if all the necessary persons come together. As an event begins, the phone adapts its ringtone settings according to the type of event. For example, the phone is muted during a conference. As the event ends, the phone's normal mode of operation is restored. Since Glaubitt's work is not based on spatial ontologies, locations can only be specified in geographic coordinates, which narrows the usability of the system. It has also not yet implemented street map data for route planning, so that the distances are only rough approximations.

For our planner, we use commercial Web-services for routing and geocoding provided by PTV[1]. The *eLocate* [PTV AG, 2004a] server converts postal address data into geographic coordinates so that the addresses can then be invoked for route calculations. The postal address can be incomplete and can also contain spelling mistakes. The *eRoute* [PTV AG, 2004b] server is based on an algorithm calculating the shortest routes on the basis of cost, using a digital road network. The costs per trace consist of the costs per time unit and the costs per route. The speed for a trace is determined by the type of road and the vehicle profile for this type. The algorithm can choose between the shortest route, the quickest route, or a linear combination of the two.

For the exact calculation of walking distances, we use the Yamamoto toolkit [Stahl and Haupert, 2005] that allows us to model multi-level indoor environments and supports route finding for pedestrians. In [Stahl and Heckmann, 2004] we describe a hybrid location model that consists of a semantically supported symbolic location model as part of the UbisWorld spatial ontology (see www.ubisworld.org) and the geometric location model of the Yamamoto toolkit. Aside from its core functionality for navigation, it also provides a common notion of location to mobile applications. This requires a standardized and structured vocabulary referring to real-world locations that allows for spatially indexed information and the exchange of positions between applications. Our model represents real-world places at different granularities as resources in the World Wide Web. The geometrical model is joined with UbisWorld by uniform resource identifiers (URIs) as location identifiers.

In order to realize the location-adaptation of our system, we need knowledge about the current whereabouts of the user. If the user is staying outside, this can be done

[1] PTV Planung Transport Verkehr AG, Karlsruhe. Website: http://www.ptv.de

with the use of GPS. Unfortunately, GPS is not working in indoor environments, e.g. a shopping mall or an airport. Currently there are several systems under development to overcome this restriction. 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 first case, the so-called exocentric localization, the user is sending information to the environment and some centralized server uses this data to calculate their position. In other words, the user is tracked. In the latter case, the egocentric localization, the user receives information from the environment and their personal device uses the data to calculate the current position. In [Brandherm and Schwartz, 2005] and [Schwartz et al., 2005] we describe the basics of an egocentric localization system that uses geo referenced dynamic Bayesian Networks to determine the position of a user. This system, which is now called *LORIOT* (Location and ORientation in Indoor and Outdoor environmenTs), uses infrared beacons (IR beacons) and active RFID tags to determine the user's position. LORIOT runs on a PDA and uses the built-in infrared receiver and an attached active RFID reader as sensors. Each IR beacon sends out a 16 bit wide identification code. Receiving such a beacon gives a 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 the PDA's IR sensor must be in the line of sight of the beacon and can thus very easily be blocked. The RFID tags on the other hand send their information via radio waves, which can be received even when the PDA resides in the user's pocket. But due 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 is missing a direction-information. By combining these two sensor readings with the help of dynamic Bayesian Networks, we achieve what we call an Always Best Positioned system (ABP system): As long as there are either IR beacons or RFID tags detectable, we will be able to estimate a position whose precision depends on the type of the sender. If we can receive both, we will get an even higher precision. In Section "Localization" we describe a system that can be used to estimate positions in an uninstrumented environment.

3 A Web-based Task Planner

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. In our context, each task acts 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. Initially, the task is not assigned and the checkbox is greyed (disabled). Once the task is assigned to someone, the checkbox becomes active (enabled). Finally, the task can be checked by the responsible user to indicate completion or marked with a warning sign to indicate failure. In analogy to a calendar, each task can also represent an event.

The user can choose its type between a fixed date and an open interval. The prior begins and ends exactly at the given dates, whereas the latter type describes a task more loosely by its earliest start and/or latest finish (deadline). In order to support coordination, multiple users can be invited and associated with tasks. Collaborative tasks are also supported as documents and references to websites can be attached to shared tasks.

The *Ubidoo* task planner is running on a Web-server[2] and can be accessed everywhere via the internet. The user interface for desktop PCs is split in three frames, as shown in Figure 1. On the left hand side, the users' tasks and subtasks are shown in a foldable tree structure. 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.

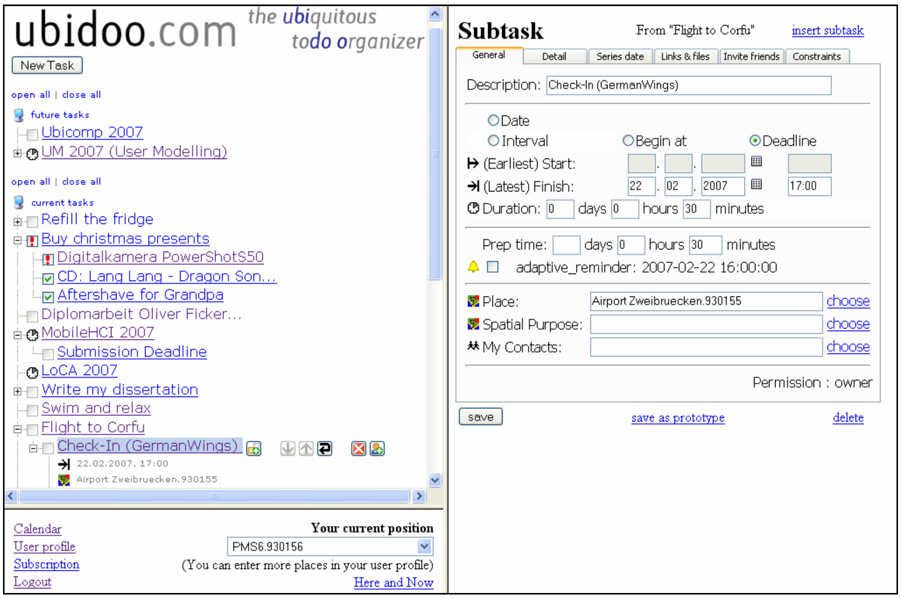

Figure 1: The Web-based user interface for the ubiquitous task planner *Ubidoo*.

Adaptive task reminder

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

[2] The **ubi**quitous to-**do** **o**rganizer. Website: http://www.ubidoo.com/

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 to dress accordingly or to 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 we can't 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 Figure 1). As the location of the user (according to the user profile) changes the task planner updates the distance between the user and all events using route planning web services. The PTV *eRoute* service returns the estimated driving time between two locations. In addition, we consider the walking distances to the next parking lot or bus stop. The concept of web services makes it relatively easy to include other services in our planner. Semanic web technologies like WSDL (Web service description language) and SOAP (Simple object access protocol) let us send the same requests for routes to a web service for public transport systems, such as busses and trains. The planner would than choose between car and bus based on the user's preferences.

For the sake of simplicity, the actual reminder functionality has been implemented by sending an email to the user that includes the description of the event and a link to the task planner. This method requires a mobile device with an email client, but one could basically also use an SMS gateway.

Here-and-now: an adaptive task view

We have seen how a small improvement like the location-aware reminder can ease the use of calendars and alarms in the context of mobile devices, but even such a reactive calendar is still far from the goal of a 'personal assistant'.

By adding more general knowledge about the relationship between tasks, activities and locations through the use of ontologies, we can make further progress towards this goal. The 'here-and-now' view filters the user's tasks according to the current situation, which depends on time and location of the user, so that it suggests only those tasks which can actually be worked on. The time horizon of this view is limited by the next binding event to occur (with a fixed beginning or deadline), and to be more precise, by the time of this event's adaptive reminder.

Regarding location, the adaptive view does not simply match the location of the task with the user; similar to the adaptive reminder, the system performs a route calculation to estimate the actual distance and time to get there. Furthermore, the system considers the opening hours of a place, so that the planner can filter those locations which are currently closed. In combination with route knowledge, it can even find out if the location will close before the user can actually get there.

Likewise, the system considers the purpose of locations in terms of activities on a semantic level. If the user associates a task in the planner with activities from the ontology such as *shopping for electronics* instead of a certain location instance or 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 2 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 kilometers (or 11 minutes by car) away from the current location of the user (in 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 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 minutes 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 travelling time is now more than 4 hours.

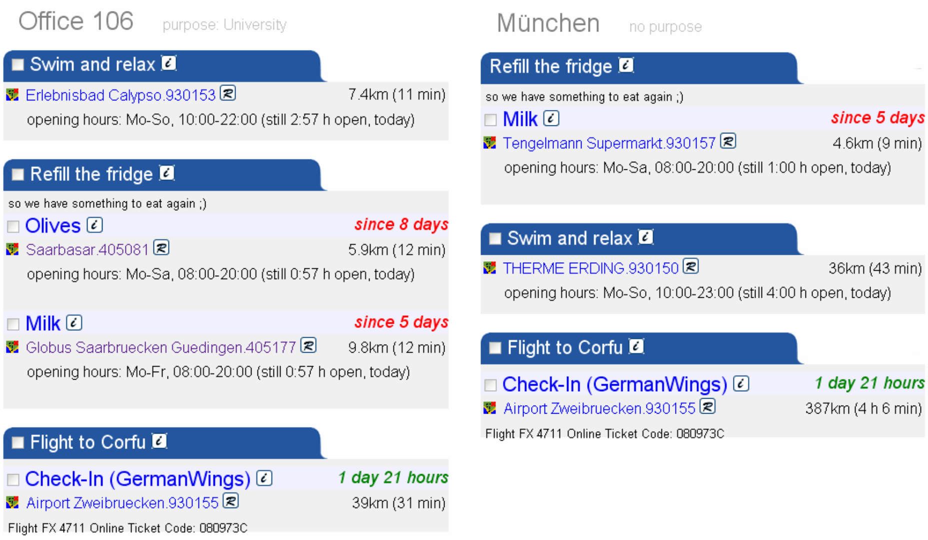

Figure 2: The location-adaptive task view as seen in Saarbrücken (left) and Munich (right).

Altogether, the adaptive view has several benefits for the user. It generally reduces the user's cognitive load that is required to browse through the complete task tree (equivalent to a calendar and to-do list). On mobile devices, it helps to save valuable screen space on the limited displays. Likewise, it helps to proactively present only relevant information to the user on public displays, which often do not allow for interaction (no touchscreen or mounted out of reach to prevent vandalism). We have developed a public display infrastructure called *IPlay Blue* [Schöttle, 2006] for the task planner that uses non-interactive, wall-mounted displays. The users are automatically recognized by scanning for their mobile Bluetooth devices.

4 Ubiquitous Domain Modelling

Our ubiquitous task planner is based on a centralized SQL database that includes the task descriptions of all its users. The user-adaptive and location-aware features are realized by linking user and task descriptions to the UbisWorld knowledge base. In the following we introduce UbisWorld and describe our extensions to the ontology.

User modelling

The UbisWorld knowledge-base has been designed to model and query the characteristics of a user, including their activity, as well as the environmental context. It also provides a symbolic spatial model to express location as described above. One goal of UbisWorld was to provide a flexible web-based model that can be inspected by the user through a convenient user interface and accessed by applications using semantic web technologies. In order to represent the user's current location, tasks or preferences, we use the general user model ontology GUMO (see www.gumo.org).
An important information for a navigation planer is for example the users's preference between car or public transport in certain situations. However since we do not want all systems to ask the user the same question several times, we use the new concept of ubiquitous user modeling which describes ongoing modeling and exploitation of user behavior with a variety of systems that share their user models. In our case, the user model is shared between the task planner and the public display infrastructure [Schöttle, 2006] which comprises several applications like a situated shopping list and personalized advertisements. The public displays' user interfaces are implemented through PHP scripts and Web-browser components. The ubiquitous user model provides the users' Bluetooth ID for recognition and a personal graphical icon which allows to address the user in an anonymized manner.

Location modelling

The UbisWorld ontology includes spatial concepts to represent locations in different granularities like country, city, building and room which can be seen in Figure 3. Location instances have a profile that is similar to a user profile and consists of situational statements. The spatial relation of *inclusion* is used to express the connection between rooms and buildings, so that the general attributes of a room can be inferred or inherited from the building's profile.
The profile of a building contains information about its postal address and geographic coordinates, the opening hours if applicable and the general purpose of the building in

terms of activities that can be done there. Some example purposes are depicted in the upper right of Figure 3, and an instance of a location profile is depicted in Figure 4. Further information is given about the nearest bus stop and parking lot so that additional walking distances can be estimated which are not considered by a route planner based on street maps. For fine-grained geometric modelling of pedestrian routes and buildings, we use the Yamamoto [Stahl 2006] toolkit. It allows for representing indoor environments in multiple levels and includes a routing component that provides shortest paths for pedestrians. Using the spatial *map reference* relation, location instances in UbisWorld can be linked to their geometric representation in the Yamamoto models. Likewise, the geometric building and room models include symbolic references to their counterparts in UbisWorld. In combination, UbisWorld and Yamamoto form a so-called hybrid location model as described in [Stahl and Heckmann, 2004].

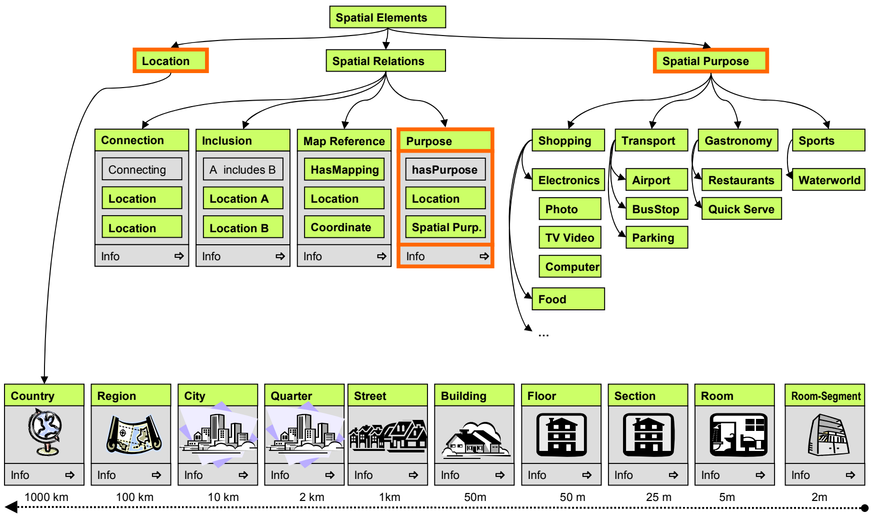

Figure 3: Spatial concepts in UbisOntology: locations, spatial relations and spatial purposes.

Task modelling

For task modelling, we use a separate *Mysql* database which uses *UbisIdentifiers* as references to elements in the UbisWorld knowledge base. UbisIdentifiers are strings that consist of a descriptive, human-readable part and a six-digit unique identifier, separated by a dot. Each task can be associated with a parent task and one or many users of the UbisWorld. Tasks can be localized either directly by referencing spatial elements within UbisWorld (e.g. `Building E11.400040`), or indirectly by choosing them in terms of activities from the spatial purposes provided by the ontology (e.g. `Education.300060`). Through these references, the task planner can retrieve the address details and opening times for locations from UbisWorld's location model.

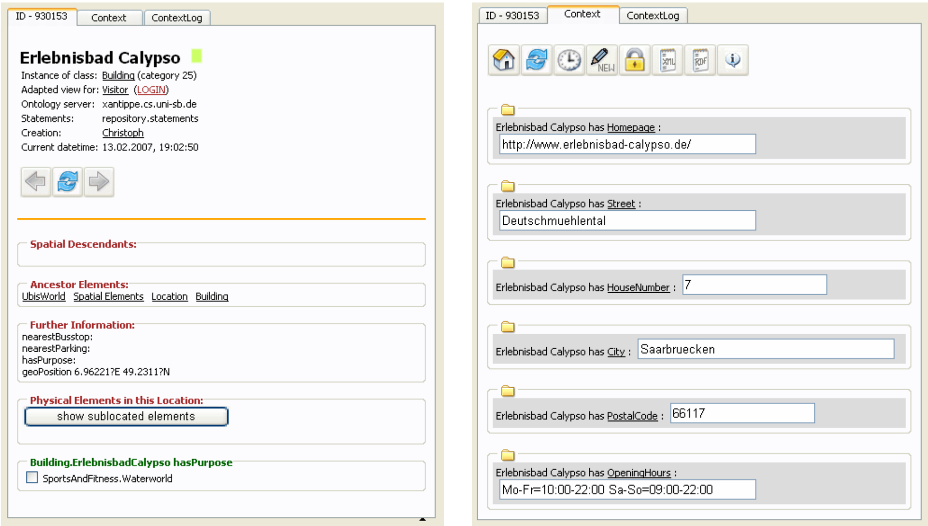

Figure 4: The location profile for a waterworld called Calypso in UbisWorld.

5 Localization

As stated in the Related Work section, we can use LORIOT to determine the user's position. Since LORIOT is an egocentric positioning system, the user can give permission to the system to send the current position to UbisWorld, where his user model will be updated accordingly. The disadvantage of LORIOT is that the user has to wear an additional device (a PDA) that is furthermore extended by non-standard hardware (the active RFID reader). To overcome this disadvantage we thought of another way to determine the user's position, which can be used in conjunction with public displays and the mobile phone of the user: We equipped all of our public displays with standard Bluetooth dongles. A small program is running in the background of each display that constantly scans its vicinity for active Bluetooth devices, e.g. mobile phones. Every Bluetooth device has its own unique address and a user can choose to store this address in his personal user model in UbisWorld. Each display knows about its own location and by a lookup of the received Bluetooth addresses in UbisWorld, it can infer which users are standing in front of it. According to the definition in the Related Work section, this is an exocentric localization method. Its precision is very coarse and depends on the range of the used Bluetooth dongles.

To extend the idea of the Always Best Positioned paradigm, we are currently researching on a combination of standard sensors that can be found in state-of-the-art mobile phones: GSM radio cells, Bluetooth and WiFi. The information to which radio cell a mobile phone is currently connected to, can give a very rough estimation of the location of a user. For example, the campus of Saarland University is completely covered by one cell. Detecting that cell gives evidence that the user is currently on the

campus. Also there are a lot of different WiFi networks set up on the campus, so that detecting our laboratory's own WiFi network can refine the estimated position to "somewhere near or in the lab". With all the Bluetooth enabled displays and computers in the lab, detecting certain stationary Bluetooth devices can give evidence on a room level. The number of the detected Bluetooth devices can also be used to infer the situational context of the user. A very high number of detected mobile phones can give evidence that the user is currently having lunch in the refectory, a low number that he is in a meeting and an increased number that he is visiting or giving a lecture.
We plan to use machine-learning methods to accomplish this, so that every user can train the system to their own needs. An application on the mobile phone will constantly read all its sensors and guess the current location. If the user wants to use the location-based task planner, the application will propose the estimated location (or no location at all, if the system is completely untrained). The user can then refine this proposal, where the possible semantic descriptions of the location are taken from UbisWorld. The application will then try to attach this refined location to its current sensor readings, where it can use the hierarchical spatial ontology of UbisWorld to assign the different layers of the ontology to the different precisions of the sensors.

6 Outlook and Conclusions

We have presented a web-based ubiquitous task planner that combines a calendar with a to-do list. Each task can be associated with a location or a general activity using the UbisWorld ontology and knowledge base. We employ a route planning web service to estimate geographical distances based on street maps so that the planner can adapt its reminder functionality and task view to the current location of the user. Semantic knowledge allows the planner to automatically suggest alternative places. We have adopted a pure reactive approach for our system, since we believe there is too much uncertainty involved in everyday life for a reasonable planning approach.
The mobile internet is still expensive to use today, and it might seem like a disadvantage to store ones calendar on a server instead of the mobile device itself. Also privacy issues might scare people off from using such a service. On the other hand, web-based email services are very popular and preferred by many users against local mail clients for the sake of ubiquitous access and less configuration. We believe that the mobile internet will become affordable soon through flat rates and that server-based PIM solutions will gain popularity, since they are device independent and avoid synchronization problems, which are quite common today. Another major advantage is the easy sharing of tasks and events between friends and colleagues.
In the context of user modelling, our task planner has demonstrated how applications can gain intelligent behaviour through adaptation based on a ubiquitous user- and location model. The UbisWorld web interface provides the necessary building blocks to choose locations and activities from the ontology by a few clicks. However, the current implementation of the selection tree does not scale with the size of the ontology and needs to be improved through *AJAX* Web technology.
For the future, we will use the ubiquitous task planer as a platform for more specialized user assistance applications in ubiquitous computing scenarios. We are

going to integrate the task planner into an intelligent kitchen environment to serve as a shopping aid. As a user drops something in the augmented waste basket, the item will be recognized through *RFID* technology and will be automatically added to the shopping list so that it appears in the supermarket-adapted viewing context.
Further research is needed to shed light on the lifecycle and dependencies between tasks in the to-do list and events in the calendar. It is typical that collaborative tasks become events, for example if we make an agreement to go shopping after work. On the other hand, calendar events like a birthday party might spawn new tasks in the to-do list like deciding and shopping for a present. Such transitions are currently not well supported by the task planner's user interface and need to be studied in more detail.

References

1. [Brandherm and Schwartz, 2005] Brandherm, B. and Schwartz, T. (2005). Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems. First International Workshop on Location- and Context-Awareness. In: T. Strang and C. Linnhoff-Popien (Eds.): LoCA 2005, LNCS 3479, pp. 223 – 234, 2005. Springer-Verlag Berlin Heidelberg.
2. [Fensel et al., 2003] Fensel, D., Liebermann, J., and Wahlster, W., editors (2003). Spinning the Semantic Web. Bringing the World Wide Web to Its Full Potential. MIT Press, Boston.
3. [Glaubitt, 2006] Glaubitt, U. 2006. Realisierung eines situationsabhängigen und reaktiven Kalenders für mobile Benutzer. Bachelorarbeit an der Technischen Universität Darmstadt, Fachbereich Informatik, Fachgebiet Telekooperation, Prof. Dr. Max Mühlhäuser.
4. [PTV AG, 2004a] PTV Planung Transport Verkehr AG (2004). eLocateServer 3.0 Documentation. A server for geocoding address data and its request object. PTV AG, Karlsruhe.
5. [PTV AG, 2004b] PTV Planung Transport Verkehr AG, (2004). eRouteServer 3.0 Documentation. A server for calculating routes and its request object. PTV AG, Karlsruhe.
6. [Schöttle, 2006] Schöttle, F. (2006). IPlay Blue: Interaktive Displays mit Bluetoothbasierter Benutzererkennung. Fortgeschrittenenpraktikum am Lehrstuhl für Künstliche Intelligenz, Prof. Wolfgang Wahlster, Universität des Saarlandes.
7. [Schwartz et al., 2005] Schwartz, T., Brandherm, B. and Heckmann, D. (2005). Calculation of the User-Direction in an Always Best Positioned Mobile Localization System. Proceedings of the International Workshop on Artificial Intelligence in Mobile Systems (AIMS 2005), Salzburg, Austria, 2005
8. [Stahl and Haupert, 2006] Stahl, C. and Haupert, J. 2006. Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Environments. 2nd International Workshop on Location- and Context-Awareness. In: M. Hazas, J. Krumm, and T. Strang (Eds.): LoCA 2006, LNCS 3987, pp. 74 – 85, 2006. Springer-Verlag Berlin Heidelberg.
9. [Stahl and Heckmann, 2004] Stahl, C. and Heckmann, D. (2004). Using semantic web technology for ubiquitous location and situation modeling. The Journal of Geographic Information Sciences CPGIS: Berkeley, 10(2):157–165.
10. [Studer et al., 2003] Studer, R., Hotho, A., Stumme, G., and Volz, R. (2003). Semantic web - state of the art and future directions. Künstliche Intelligenz, 3(3):5–9.