Modeling and Learning Relevant Locations for a Mobile Semantic Desktop Application

Wolfgang Woerndl[1], Florian Schulze[1], Valentina Yordanova[2]
[1] Technische Universitaet Muenchen, Germany
{woerndl,schulze}@in.tum.de
[2] Ludwig-Maximilians-Universitaet Muenchen, Germany
yordanov@informatik.uni-muenchen.de

ABSTRACT: *"SeMoDesk" is an approach to implement the Semantic Desktop on mobile devices. The idea is to allow users to manage their personal information space using personal ontologies. In this paper, we are presenting our solution to improve location-awareness in this scenario. We have designed a location and sensor ontology as an extension to the personal ontology. This ontology is then used to retrieve relevant resources according to the current user context. For this purpose we have designed a resource recommendation function that is utilizing the ontology graph to find other related resources such as persons or documents. We describe the design, implementation and test of an extended SeMoDesk with a focus on the integration of a RFID infrastructure for indoor location-awareness. Furthermore, we have implemented a method to display current resources and points-of-interests in the user vicinity on a map. To learn relevant user locations, we have designed and implemented a solution based on a time-based clustering algorithm. This method has been evaluated with good results in our scenario.*

Keywords: Desktop; Mobile; Context; Location-awareness; Ontology; Clustering

Received: 2 December 2009, Revised 9 January 2010, Accepted 19 January 2010

1. Introduction

The acquisition, maintenance, retrieval and sharing of personal information such as contact data, bookmarks or appointments is becoming more and more important in our networked world. One approach to support such activities is the "Semantic Desktop" [Sauermann et.al. 2005]. The main idea is to allow users to manage their personal information space using Semantic Web technologies and ontologies. While most of current research and applications focus on desktop scenarios, we have developed "SeMoDesk" which is an implementation of the Semantic Desktop idea for mobile devices [Woerndl and Woehrl 2008, Woerndl and Hristov 2009]. Organizing personal information on mobile devices is even more difficult. This is mostly due to the fact that mobile devices have limitations in network bandwidth, storage capacities, displays and input capabilities.

In addition, the key point of supporting the user with mobile technologies is context- and especially location-awareness. A mobile device is usually used in different contexts (e.g. work or private) and in very different locations, in contrast to a desktop computer which is often exclusively used in the user's office only. Therefore, it is important to adapt the management of personal information to the current user position. Consider the following scenario: A user enters a lab or meeting room with her mobile device. The system finds all related documents based on the current location, for example a presentation that is related to an upcoming meeting in the room.

In this paper we explain our approach for modeling and learning relevant locations in the mobile Semantic Desktop solution SeMoDesk. The rest of this paper is organized as follows. First, we give some background on the Semantic Desktop and SeMoDesk in Section 2. In Section 3 we then explain how to model spatial entities and introduce location and sensors in the personal ontology. In Section 4, we describe the design, implementation and test of an extended SeMoDesk with a focus on the integration of a RFID infrastructure for indoor location-awareness. In Section 5, we briefly explain another part of our work that supports the user by displaying resources on a map using GPS for outdoor location-awareness. In Section 6, we describe our solution to automatically learn user locations that are relevant to a user. Finally, we conclude the paper with a short discussion of related work, a short summary and outlook in Section 7.

2. Background

2.1 The Semantic Desktop and PIMO

The Semantic Desktop is an approach to integrate desktop applications and the data managed on desktop computers using Semantic Web technologies [Sauermann et.al. 2005]. Thereby, ontologies are used to formalize relationships between resources and define a concept hierarchy. A user can assign meta data to all data objects that she uses on her computer. Relations between resources can be defined with the goal to integrate desktop applications and enhance finding relevant information. For the Gnowsis project, the “Personal Information Model” (PIMO) ontology was designed [Sauermann et.al. 2007]. We have based our application on the PIMO ontology. The overall goal of PIMO is to define a concept hierarchy allowing a single user to formulate her view on tasks, contacts, projects, files and other resources.

In PIMO, one basic idea is to distinguish between “Thing” and “ResourceManifestation”. “Thing” is a superclass of abstract concepts and physical objects, with the goal of representing them on a conceptual level. “ResourceManifestation” is a class to represent the actual documents on a computer system [Sauermann et.al. 2007]. All objects in PIMO can be connected to each other using relationships.

2.2 SeMoDesk – A Mobile Semantic Desktop Application

While there are Semantic Desktop implementations and related systems like the aforementioned Gnowsis available for desktop computer use, there is little for mobile environments. Therefore, we have designed and implemented SeMoDesk, which is a realization of the Semantic Desktop idea for PDAs [Woerndl and Woehrl 2008]. The main design goals were to account for the restricted resources of PDAs, to build a stand-alone application (i.e. not a client of a Semantic Desktop server solution), because of possible network limitations, and adaptation to and usability on the mobile device. For example, phone calls and SMS messages are integrated which is not the case in related, desktop based approaches. To do so, SeMoDesk assists as much as possible, for example calls or short messages on the mobile device are integrated automatically.

The classes and instances of the personal ontology can be browsed in SeMoDesk (Fig. 1, left). For example, all messages or calls with one person, or all resources such as documents or appointments that are associated with a project, can be displayed with one tap on the touchscreen of the mobile device (Fig. 1, right). However, browsing the ontology is not enough, as the following example illustrates. If a user is in a meeting right now, she might not only be interested in documents that are directly related to this meeting, but also messages that are related to a project or a person that is related to the meeting, or contacts that are concerned with a relevant topic, and so on. The goal of this work was to design and realize this kind of recommendation method which will be explained in the next subsection.

2.3 Recommending Resources in SeMoDesk

Users can use a context-aware recommendation function to improve information access and find relevant resources [Stahl and Heckmann 2004]. The recommender consists of the following two steps:

1. Finding current resources (Fig. 2, left), i.e. resources that are of interest for the user right now

2. Recommending other items, starting from the instances found in step one

In step one, users have the option to manually select concepts or resources. In addition, the system proposes items in this first step based on the current context: date, time and location. As a result, the system displays a list of resources which are

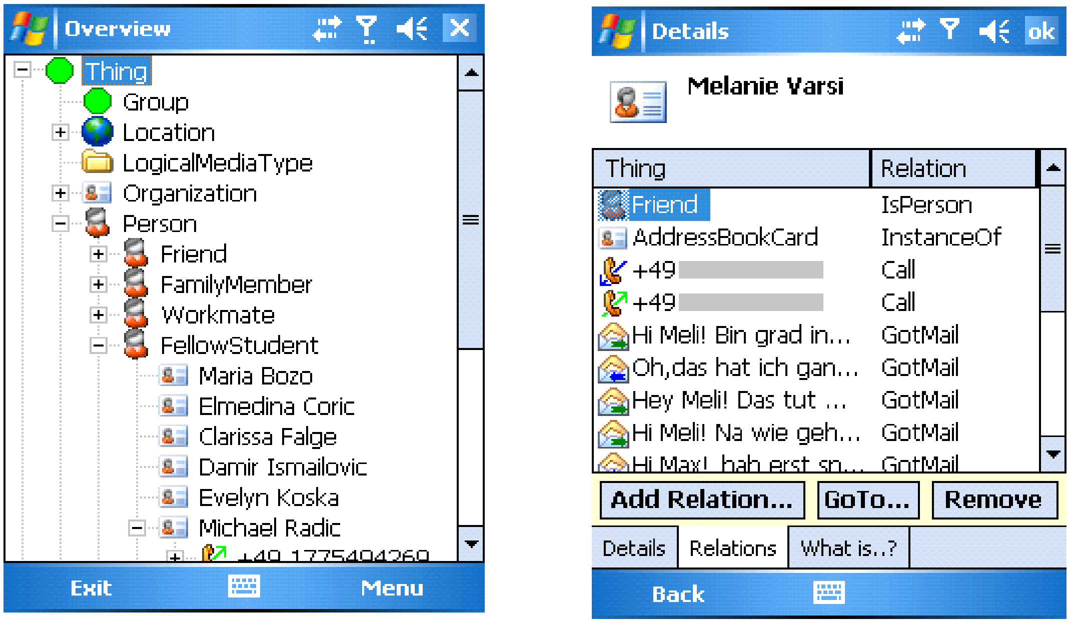

Figure 1. Browsing the personal ontology (left), and displaying existing relations for an instance (right)

Figure 2. Finding current resources (left), and displaying the relation path (right)

of current interest to the user. In Fig. 2, left, a list of resources in a certain timeframe is shown (button "On Schedule"). This list may already contain relevant resources, but these elements mainly serve as possible starting nodes for the more advanced search in step two. The user can select one node and start the recommendation process: the system traverses the graph of concepts and resources and analyzes every node according to an evaluation function using configurable heuristics. For example, if the recommender finds a path between the starting node and a particular project resource via "related" edges, this project is considered relevant in the current context. The details of this recommendation process are explained in [Woerndl and Hristov 2009]. As a result, the system displays several resources and also information on why these results are suggested, i.e. the path from the starting node to the result entry (Fig. 2, right).

3. Modeling Spatial Entities in a Ubiquitous Environment

The first version of SeMoDesk allowed users to manage, browse and query their ontology. The focus in this paper is on how to introduce location awareness in the mobile Semantic Desktop approach. To do so, the first step is to model location in the personal ontology.

3.1 Existing Related Approaches

Modeling spatial entities has a key role in location-awareness. The recent research literature comprises several projects aimed at creating suitable location models considering different needs and approaches, three of which outline in this subsection.

UbisWorld is a ubiquitous symbolic location model developed by a research team at the Saarland University [Stahl and Heckmann 2004]. The overlying project REAL aims at supporting users navigate in sensor-equipped (RFID and infrared) shopping environment. Location information is processed with dynamic Bayesian networks that apply common Bayesian filters to stabilize positions in spite of bad sensor readings. A so-called geo-referenced dynamic Bayesian network is used to overcome the necessity for building a grid and therefore enable the calculation of a user's position on his own small hand-held device without a connection to an external server. The ontology used to model the entities is based on a hybrid system to bridge the gap between the coordinate based pedestrian positioning system, i.e. GPS, and the symbolic ubiquitous world model [Stahl and Heckmann 2004].

The Middle Building Ontology (MIBO) was introduced by [Kay et.al. 2007] because former location models did not provide a sufficient basis for modeling indoor places. Thus, MIBO focuses on spatial elements inside a building. In a follow-up paper [Nui et.al. 2009] describe how MIBO's taxonomies (hasPart and isA) can be used for reasoning and conflict resolution.
The Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) is an actual set of ontologies, consisting of a core and an extension set. SOUPA uses several known ontologies like FOAF, DAML-Time, COBRA-ONT, MoGATU BDI as well as the Rei Policy Ontology, combining the expertise of these ontologies in each one's according area. The representation of locations and location context is mostly realized by the OpenCyc Spatial Ontologies and RCC. The first is used for symbolic representation of space while the latter can model geo-spatial coordinates for various types of geographical regions, as well as spatial relations between them [Chen et.al. 2004].

3.2 Modeling Location in the Personal Ontology

To accurately model a user's position, a system requires a detailed location model where even small places can be distinguished, especially when it comes to indoor tracking. On the other hand, a hierarchic design with common "super classes" (representing composed objects) is needed for reasoning. This ambivalence results in the demand for a diversified granularity model.
As we worked on an RFID-based positioning system, our ontology allows in-depth modeling of a building's floor plan. Still, we also had to take other technologies into account to permit future work on the system. Following several respective ontologies for indoor and outdoor tracking, these classes were found adequate to fit our needs:

- Country
- City
- Area
- Street
- Place
- Building
- PartOfBuilding
- Level
- Floor
- Room
- PartOfRoom

This wide range of granular diversity allows us to model huge areas as well as small spots inside a building. Object level was not considered for this project as we want to model actual places.

The location taxonomy can be implemented by using HasPart – IsPartOf relations. Additional relation types like IsNearby or IsOnThe-RightSideOf/IsOnTheLeftSideOf can easily be added in the future.

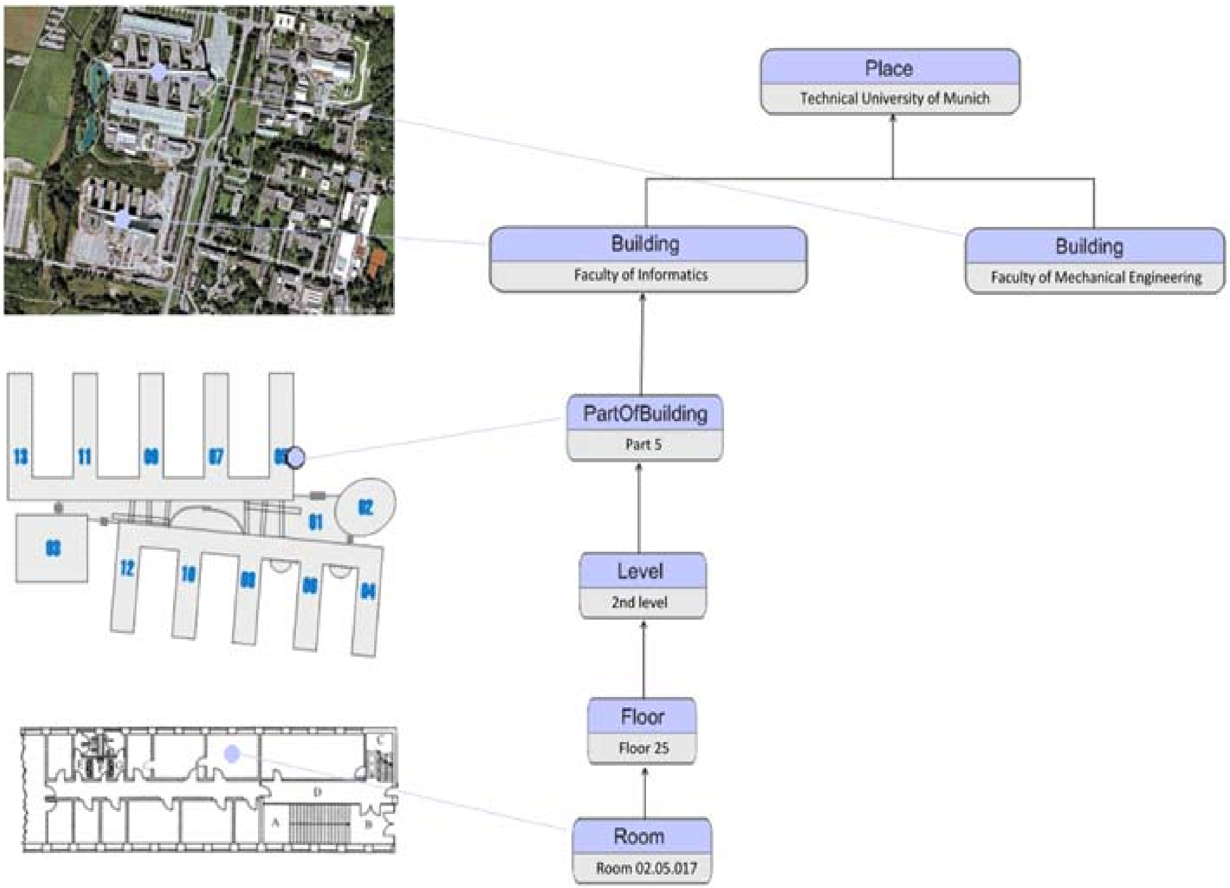

Figure 3. Sample model instance

Due to the taxonomy, evidence conflicts caused by granularity variance (e.g. a low-range Bluetooth sensor within the range of a high-range WiFi sensor, overlapping sensor ranges in general) can easily be resolved. [Niu et.al. 2009] describes an algorithm called Closest Common Subsumer that can be implemented. The algorithm uses the location ontology to find the closest common subsuming location: this is similar to the lowest common ancestor concept in graph theory. When a user is spotted by the WiFi sensor covering the upper level of a building and also by the RFID sensor in room 02.05.017, the reasoner can check the relation graph of both locations. As room 02.05.017 is actually ON the upper level and thus on the direct hierarchical path in the taxonomy, there is no conflict.

Figure 3 shows a sample model instance of our university with IsPartOf relations. Every spatial entity can be modeled, from the campus as a whole to every single workplace in the faculty buildings.

3.1 Modeling Sensors and Identification Tags

Subject to the technology used for location spotting or tracking, we need to model sensors and receivers respectively identification tags. While some technologies require the addressing of an external (in most cases also stationary, e.g. RFID) sensor, others offer built-in sensors. The former is accompanied by the fact that the addressing requires the system's knowledge about the sensor's position. Furthermore the sensor itself has to be modeled as an entity, its position and its address (for establishing a contact) assigned to it. That again means that both of these attributes have to be modeled as entities and set in relation to the sensor. .

We have designed a superclass "SensorTags" and subclasses, depending on the technologies used for positioning:

- Sensor: Represents a stationary sensor (e.g. RFID)

- IP: IP entities are linked to Sensor entities and hold an attribute with the IP address to reach a specific sensor on a network (e.g. WLAN).

- RFID Tag: RFID tags represent the devices or objects they are linked to in the ontology. They hold a numerical ID.

- Bluetooth Tag: Bluetooth combines sender and receiver in a single unit that is built-in in electronic devices. Unique numerical tags identify those units.

-GPS Coordinates: Entities of this class hold coordinates for a certain place and are linked to either a Place or a Building entity. A user is identified by a device with an RFID tag in our model. For reasons of simplicity we decided not to separate user and device positioning. All sensor tags shall be set in direct relation to the user whose devices they are attached to. However, this entails a problem: possible evidence conflicts. If a user owns two devices and leaves one of them on her desk, carrying the other one at hand, a position scan is likely to produce two positive readings, one of them a false positive. If users and devices were modeled separately, a reasoner might resolve this conflict by considering the relations of the devices to the user if those relations held priority information (e.g. "ownsMainDevice" vs. "ownsSecondaryDevice").

4. Integration of an RFID Infrastructure for Indoor Location-Awareness

After integrating spatial entities into the personal ontology, we will now describe the integration of location-awareness and a RFID infrastructure in SeMoDesk [Schulze 2009].

4.1 Architecture of the Application

For our project we were bound to a stationary RFID reader that was not physically connected to the mobile phone running SeMoDesk. Therefore, we had to adapt our design to the circumstances: A virtual connection to the reader was needed, which required the use of a split client-server architecture to manage this connection. Our development and testing environment consisted of a single reader, but in a different environment more readers make sense in order to cover an entire building, not just a single room. Thus, every server and every client needed to be able to communicate with several of its counterparts.
First off, we had to make a choice for a virtual connection between the RFID reader and the mobile device. Many state-of-the art mobile devices come equipped with WLAN and Bluetooth units. In comparison, WLAN can cover a much bigger area in terms of operating distance, which predestines this technology for our location system. Additionally, we let the wireless network handle authentication by using a WPA2-secured WLAN connection instead of handshaking procedures in the system. As devices usually only have a single WLAN interface, a connection has to be of short duration in order to approximate a simultaneous communication between multiple entities. Therefore, connections are not kept alive but are instantly terminated after a query.

The server part is designed to be a stand-alone application handling both the virtual connection to our device and the physical connection to the RFID reader. It listens for any incoming query sent from SeMoDesk's client part to return a list of devices in range of the reader.

The front-end had to empower the user to choose what sensor(s) to query. A simple list of sensors without their locations doesn't suffice here, as the user wants to know the contacts near a certain place. Thus all sensors are listed according to their position in form of a tree. When initiating a query a set of sensors can be picked if under the same superclass in the taxonomy. All corresponding servers are contacted, the readings are displayed in another window which also lists any Thing subclasses (files, dates, tasks etc.) related to the contacts found in the previous query.

4.2 Extending the Recommendation with Location-Awareness

The button "In Area" (Fig. 2, left) is the user interface to search for a certain area. This integrates our location system with the resource recommender that was explained in Section 2.3. Hence, the recommendation system is extended with location awareness. When the user clicks on "In Area" (Fig. 2, left), a list of currently available sensors is shown to user. The user can then select one of them, for example a sensor assigned to a meeting room she is interested in right now. What the system does is provide a list of tags, a list of persons in a certain area and also a list of all the resources related to those persons.

For the mobile device to find out its own position, an RFID tag is physically attached to the device running SeMoDesk and is assigned a client. Whenever a sensor returns that tag the system can determine its position by looking up the sensor's location in the ontology, recommending items subject to that location. Once the position is detected, the scenarios like the one mentioned in the introduction can be realized using the resource recommender. Although not currently implemented as a "one-click" function, this process can be accomplished using only the means of the framework we are providing.

4.3 Design and Implementation of the Extended SeMoDesk Application

Because of the limitations of the mobile device, we implemented the ontology using a SQL database that is available for Windows Mobile. Every class is represented by an entry in a database table THINGS. We designed a new superclass "Location" to support the location ontology. Every class of the location ontology such as "City" or "Room" is then an entry with an additional entry in RELATION to model the relationship to the other classes, e.g. a sub-class relationship. In addition, every level of the location hierarchy has been assigned an icon (see below Fig. 5, right). To model the location of instances, new relation types were introduced. "Holds" – "IsSituatedIn" serves to link a sensor and the location it is placed at. To assign IPs to a sensor, "Addresses" – "IsAddressedBy" are used. "Identifies" – "IsIdentifiedBy" constitutes a relation among an RFID tag and its corresponding resource. Finally, the location taxonomy can be implemented by using the already mentioned "HasPart" – "isPartOf" relations. These relations are also stored in a database table.

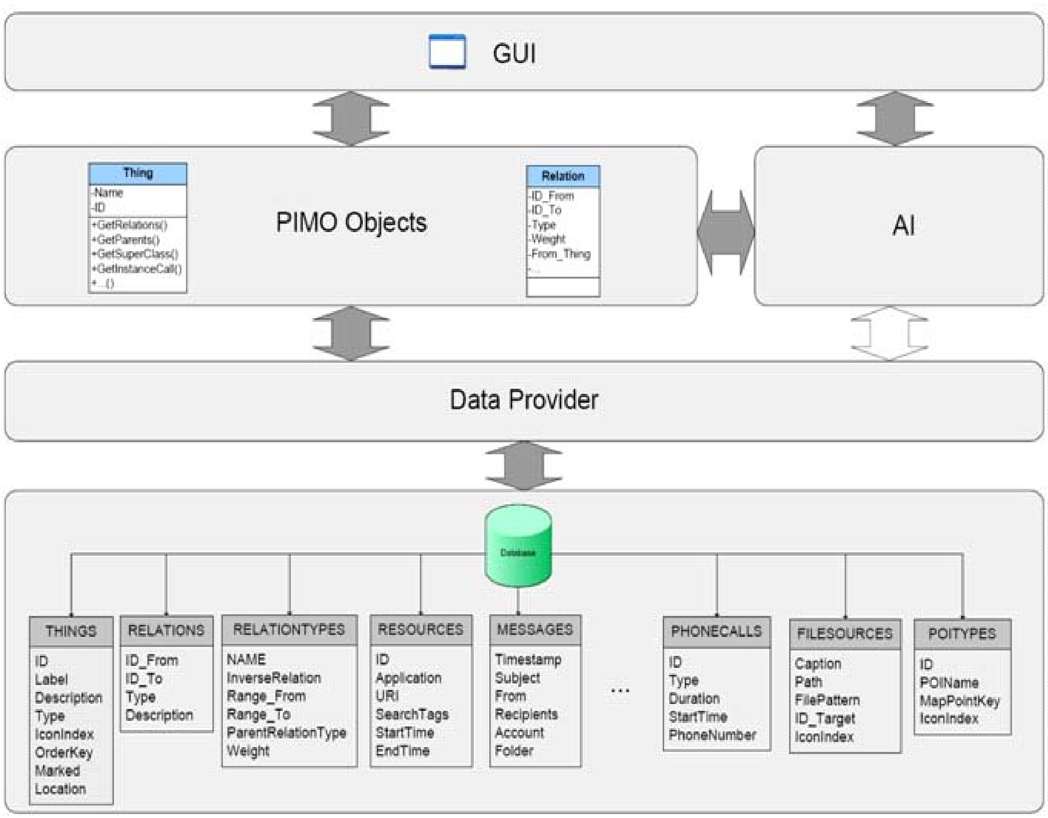

Figure 4. Overview of the system design

Figure 4 gives an overview over the main layers of the system design [Woerndl and Hristov 2009]. The main parts of SeMoDesk are components needed for the graphical user interface (GUI), a representation of the PIMO objects and corresponding data provider classes for the database access. Every item in our approach has a GetRelations() method that retrieves all corresponding relations in an efficient manner, for example. The "AI" package contains all the classes of the search and recommendation algorithm as explained above.

As stated before, we had to establish a client-server architecture for integrating the RFID infrastructure. Thereby, the client side is part of the SeMoDesk application and represented by a RFIDClient class. This class offers a method to connect to all its stored IP addresses and to receive a list of RFID tags. The server side is connected to the RFID reader and consists of a single console application listening for incoming connection requests. It returns a list of tags in range of the sensor whenever a client connects. There is no communication aside from the transmission of the alphanumerical RFID tags. All the further procedure is done in the SeMoDesk application on the mobile device using the resource recommendation function that was explained above.

Our implementation was done using Microsoft's IDE Visual Studio 2008. The programming language is C# and the runtime environment is .NET Compact Framework 3.5. The application was tested using a HTC P3600 PDA phone and some other similar devices. SeMoDesk should run on any Windows Mobile 5 or 6 PDA with a touchscreen interface.

4.4 Test

We have tested the RFID part of the application in several scenarios using the short-range HF RFID reader Tricon Starter Kit 100.

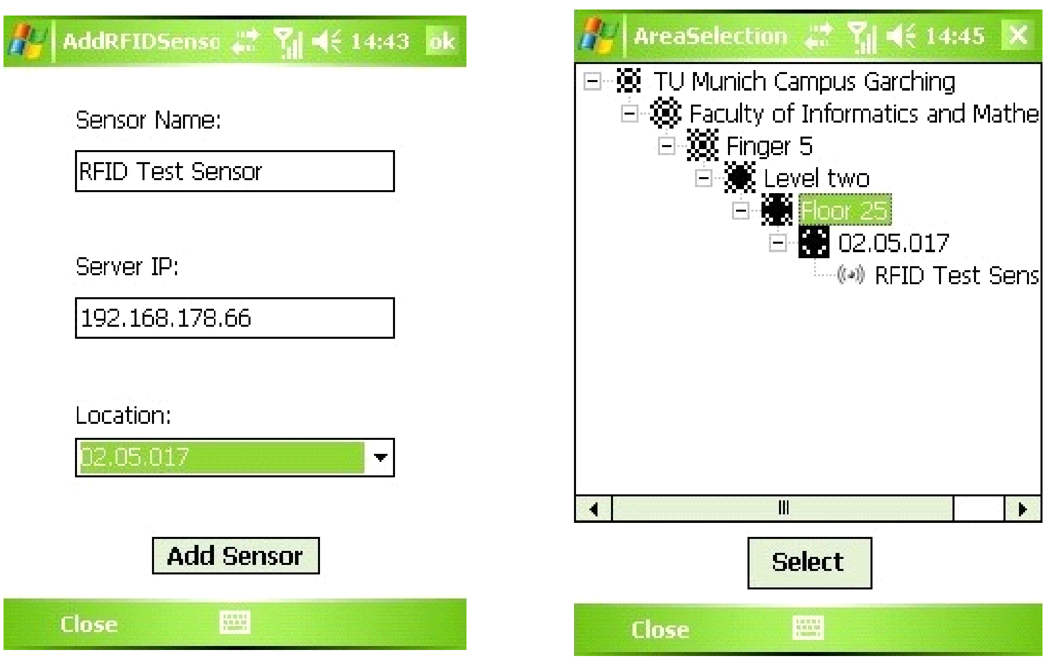

Fig. 5. Setting up a new RFID sensor (left), and selecting a floor in the location hierarchy (right)

In this subsection we explain one of those evaluation scenarios. For this purpose, we set up our ontology to model the sample hierarchy of our university as previously depicted in Fig. 3. An RFID sensor was added (Fig. 5, left) and associated with room 02.05.017.

Then, we started the location-based recommender using the aforementioned "In Area" button. In the appearing AreaSelection window (Fig, 5, right) we chose the higher location node "Floor 25" which included all sub nodes, i.e. our room with the RFID reader. SeMoDesk flawlessly connected to the server linked to that reader and received a list of tags in its range (Fig. 6, left). The one tag that was transmitted turned out to be associated with contact Diane in our ontology. By traversing the relation graph the recommender then determined all entities directly related to the contact (Fig. 6, right), namely the task "Thesis" and also the associated RFID tag.

The application proved to be working faultlessly within the scope of all conducted scenario tests.

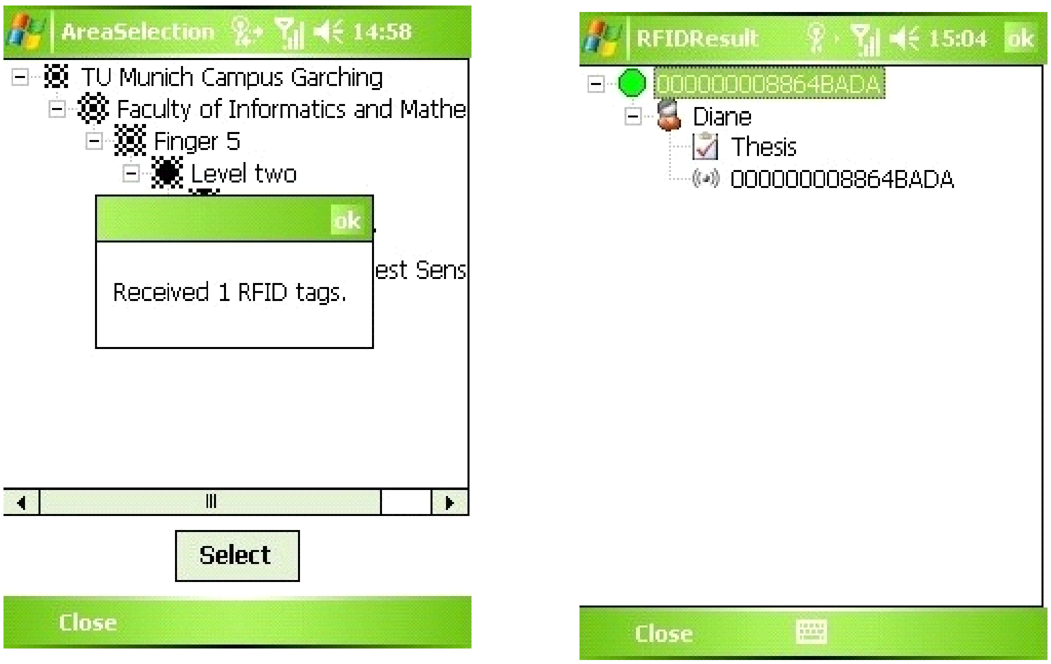

Figure 6. Receiving RFID tags (left), and displaying recommended entities (right)

5. Semomap - Displaying Resources on a Map

In addition to the integration of the RFID infrastructure for indoor positioning, we have also worked on utilizing GPS for supporting the user outside a building.

Figure 7. Associating a task with a POI concept (left), and displaying the map (right)

Until now, the explained concepts allow for searching the item space in the personal ontology. But the ontology can also be utilized to recommend additional resources, items that are not explicitly managed by the user. The application scenario is that users are looking for points-of-interests (POIs) in the current geographic vicinity to perform certain tasks. For this purpose, we have extended the PIMO ontology by a POI concept with sub concepts such as “cinema”, “restaurants”, “shop” etc. The user can then relate tasks or any other resources to POI types, as shown in Fig. 7, left. In addition, appointments (or any other resources, in theory) can be related to addresses or other elements in our location ontology as explained above. When the user starts the mapping feature of SeMoDesk, information about relevant POIs are shown on a map (Fig. 7, right), together with the location of upcoming appointments. The map is centered on the current user position. An Internet connection is required for the mapping feature to retrieve the generated map from the mapping server, in our case Microsoft MapPoint.

6. Learning Relevant User Locations

In this paper as yet, we have presented our solution to model and utilize location in our mobile personal information management application SeMoDesk. However, we have not dealt with acquisition of relevant user locations so far. User would have to enter addresses manually. This seems to be unfavorable on the mobile device due to the limitations of the user interface. Therefore, we have designed and implemented a solution to automatically learn user locations that are relevant to a user. We will present this approach in this section. The solution is based on GPS logs but similar analysis could be made with traces of locations based on indoor localization mechanisms.

6.1 Requirements and existing work

The goal of this approach is to learn relevant user locations based on GPS logs for SeMoDesk. Since SeMoDesk runs client-side on a Windows Mobile device, as explained in this article, the capturing and analysis of position data shall also run on the mobile device. Therefore, an important requirement for the considered algorithm is resource efficiency with regard to computing power and also storage capacity. Since our model for location is hierarchical, it is also important to consider location instances in various granularities. For example, we need to recognize the city “Munich” as location instance, but also street names or specific addresses including house numbers in Munich. To fulfill these requirements in our scenario, we discuss several approaches based on clustering and machine learning algorithms in this chapter.

The “comMotion” system by [Marmasse and Schmandt 2000] comMotion is a location-aware computing environment which links personal information to locations in its user’s life. For example, comMotion reminds one of her shopping list when she nears a grocery store. The system uses GPS positioning and gradually learns about the locations in its user’s daily life based on travel patterns. The system identifies a place as a region, bounded by a certain fixed radius around a point, within which GPS disappears and then reappears. However, the system does not consider the time users spent at a relevant location.

A lot of work has been performed in applying clustering algorithms to analyze spatiotemporal data. The “k-means” clustering approach by [Ashbrook and Starner 2002] for example uses a Markov model to predict where a user might go next based on a given GPS log. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is another well-known data clustering algorithm [Ester et.al. 1996]. It is a density based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. An advantage of this approach is that it is not prone to misinterpret noise and outliers. The time-based clustering approach by [Kang et.al. 2005] clusters the stream of incoming location coordinates along the time axis and drops the smaller clusters where little time is spent as irrelevant. The approach compares each incoming coordinate with previous coordinates in the current cluster; if the stream of coordinates moves away from the current cluster then the approach forms a new one. Fig. 8 illustrates this process [Kang et.al. 2005].

[Liao et al. 2007] propose a system that utilizes hierarchically structured „conditional random fields (CRF)“ to generate a consistent model of personal activities. In contrast to the other discussed approaches, this system use high level context to find locations that are relevant for a user. However, information about what a user is doing at a certain location is needed, e.g. “walking” or “working”. At last, significant user locations are derived from instances with significant activities based on frequency of occurrence and other rules.

Clustering approaches appear well suited to provide meaningful results about relevant user locations based on GPS logs. One disadvantage of k-means clustering is that the number of clusters has to be known in advance. Ideally, several test runs to determine the best number should be made which does not fit our scenario well. The time-based clustering by Kang et.al. works

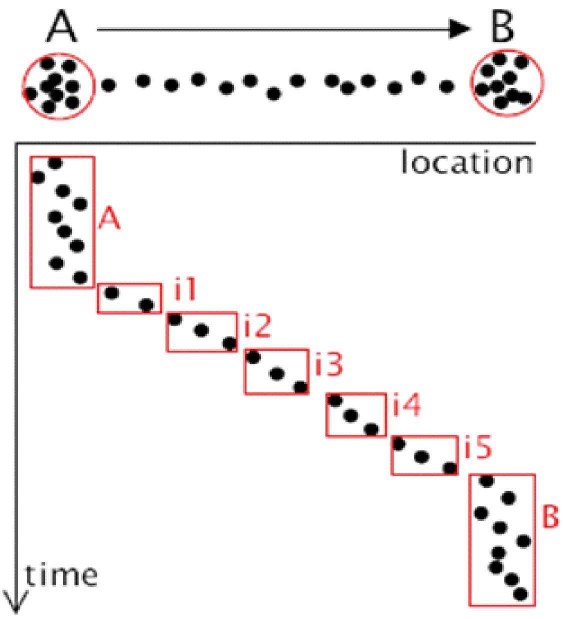

Figure 8. Time-based clustering [Kang et.al. 2005]

well even if not all data is available when the process is started. In addition, the time-based clustering is capable to identify locations that are seldom visited, but for a longer time since it incorporates the time a user spends at a location. This is not the case in the comMotion system, for example, because this system only identifies locations that have been visited at least three times. The approach by [Kang et.al. 2005] can be implemented very efficiently and is thus well suited for the mobile device. DBSCAN is less efficient with regard to time and storage complexity which is an issue on the mobile device. Therefore, we decided to base our system to learn relevant user locations on Kang's time-based clustering.

6.2 Design of our solution

To implement the learning approach, we adopted a process model consisting of the following four steps [Woerndl et.al. 2007]:

1. Acquisition of raw GPS data
2. Preprocessing and aggregation
3. Interpreting the data
4. Utilization in SeMoDesk

For steps 1 and 2, we designed and implemented a "GPS Logger" program (Fig. 9, left). We think it is important to separate the positioning from the rest of the application. By doing so, users can switch off the recording of GPS data at any time, e.g. for privacy reasons. The GPS Logger acquires the raw GPS data from the sensor which is built in the mobile device, does some preprocessing (e.g. deleting invalid entries) and produces a log file that can be used by other applications. The utilization in SeMoDesk (step 4 of the process model) has been explained above; users can associate POIs with locations and display them on a map (Section 6), for example. We will now explain more details about the most important task, the interpretation of the data (step 3).

For the interpretation, we programmed a "GPS Analyzer" application. This program inputs the GPS log file, optionally lets the user specify some parameters for the clustering, runs the interpretation and displays a list of results to the user for inclusion in SeMoDesk (Fig. 9, right). The user can then choose which addresses she wants in her ontology by selecting among the result set. The interpretation itself consists of three parts:

preprocessing (e.g. deleting invalid entries) and produces a log file that can be used by other applications. The utilization in SeMoDesk (step 4 of the process model) has been explained above; users can associate POIs with locations and display them on a map (Section 6), for example. We will now explain more details about the most important task, the interpretation of the data (step 3).

Figure 9. GPS Logger (left), and displaying a list of results (right)

- Clustering of data
- Retrieving addresses by reverse geo-coding
- Composing a hierarchy of relevant locations

For the clustering, we implemented the mentioned algorithm by [Kang et.al. 2005] on the mobile device. This algorithm produces a list of GPS coordinates – longitude and latitude, e.g. "(48.1300995333333, 11.5934058166667)". Since we do not need coordinates but named locations, we need to perform reverse geo-coding. These services translate GPS coordinates in actual place names. We selected and used the "Google Reverse Geocoder" which is part of the Google Maps API[1] family. For performance reasons, we implemented a client-side cache to store earlier geo-coding results. The GPS example above translates into "Rosenheimer Straße 6-64, 81667 Munich, Germany", for example. Hence, we gained a flat list of possibly relevant addresses. In the last interpretation step, we need to build a hierarchy of locations. To do so, our GPS Analyzer extracts place names that appear in more than one relevant address. In the example of Fig. 8 (right), the GPS Analyzer has recognized several addresses in "Munich" but only one entry in "Grasbrunn". Therefore, the locations are grouped and the user can include "Munich" in addition to the street addresses on a higher level in the ontology.

6.3 Evaluation

We evaluated our approach in a small user study with seven test users and HTC P3600 PDA phones. We configured the system to record the GPS position every 1 second. The study lasted about two weeks and took place in the German city Munich. After the interpretation of the data as explained above, the system presented the user with the determined locations and we asked them to judge the relevance of this location and whether they would include the entry in their personal ontology or not. The relevance was ranked on a scale 1-5 with 5 meaning that a location is very relevant for a user. In addition, the users were asked to record the actual addresses they were visiting during the test period. This allows for recognizing locations missed by the algorithm. Table 1 shows an excerpt of the data, the first column is the proposed location, the third the actual location as

[1] *http://code.google.com/intl/en/apis/maps/index.html*

as indicated by the users.

Location	Relevance	Actual Address	Inclusion
Augustenstr. 74-84	2	Augustenstr. 75	N
Steinickeweg	5	Steinickeweg 7	Y
Bretonischer Ring 4	5	Bretonischer Ring 8	Y

Table 1. Except from the acquired data

Due to technical problems with the GPS sensor on one device beyond our control, the data of one user was not usable. Thus we had test data of 6 users in all. We analyzed the results according to the metrics precision, recall and also “usefulness”. Precision is the measure of exactness and is defined by the number of relevant locations divided by the total number of locations our approach proposed. Recall is the measure of completeness and indicates the number of relevant locations found by our algorithm divided by the number of addresses a user visited in our scenario. A relevant location is a location the user would include in her personal ontology. This means, a determined location may not be wrong, but the user just would not deem it important, for whatever reason. In addition, we looked at the usefulness of a proposed location. This parameter utilizes the relevance score the users indicated for each address (Table 1), and weighs recall according this relevance. Table 2 shows the results from our small experiment.

	Precision	Recall	Usefulness
User 1	0,66	0,66	0,66
User 2	0,66	0,33	0,16
User 3	1,00	0,66	0,50
User 4	1,00	0,80	0,60
User 5	0,66	0,66	0,66
User 6	0,75	1,00	1,00
Total:	**0,788**	**0,685**	**0,596**

Table 2. Experimental results

Our approach based on the time-based clustering approach by [Kang et.al. 2005] detected almost 80% of the relevant locations for a user. As a baseline, we compared the results with the offline clustering algorithm DBSCAN. Applying DBSCAN results in comparable outcomes for recall and usefulness, but a significantly lower value for precision: 0,399 in comparison to 0,788.

The somewhat lower scores for recall and usefulness may be due to the fact that GPS location does not work that well in our urban test environment. The algorithm may miss a relevant location, when a user enters a building shortly after exiting a subway station, because the GPS signal is lost in this case. A possible solution is to augment the GPS positioning with other localization means, e.g. positioning based on cell-ID. In summary, our approach produced meaningful locations and worked very well on the mobile device which is one of the most important requirements in our scenario. Note that this system is intended to assist the user when entering addresses in the personal ontology, users can always reject or modify a proposed location, or manually enter additional location instances for usage in SeMoDesk.

7. Conclusion

In addition to the existing approaches regarding location ontologies (Section 3.1), there are a few Semantic Desktop implementations for desktop computer settings. The original one is the already mentioned Gnowsis system [Sauermann et.al.

2006]. Gnowsis consists of two parts, the Gnowsis server which performs the data processing, storage and interaction with native applications; and the graphical user interface (GUI) part, implemented as Swing GUI and Web-based interfaces. External applications such as Microsoft Outlook or Web browsers are integrated using standardized interfaces. There are other similar systems for personal computers or servers such as IRIS [Cheyer et.al. 2005]. However, a Semantic Desktop approach tailored towards mobile devices comparable to SeMoDesk does not exist, as far as we know.

SeMoDesk implements the Semantic Desktop idea for mobile devices, specifically Windows Mobile devices. In this paper, we have focused on our approach to improve the location-awareness of SeMoDesk. To do so, we have designed a location ontology based on existing models in the literature. This location ontology extends the basic PIMO ontology that defines a concept hierarchy for the Semantic Desktop. Since our solution is based on existing approaches and ontologies regarding the Semantic Desktop, integration with a desktop based system such as Gnowsis could be done rather easily but is out of the current scope of our work. One reasonable end-user scenario is that the user defines her personal ontology on the desktop and imports, manages and queries resources on the mobile device.

The user interface of SeMoDesk including the handling of RFID sensors is not really intended for end user deployment. Instead, it can serve as a framework on which more end user friendly applications can build on. The idea is to create specialized applications for supporting users visiting exhibitions, for example, and other more focused scenarios. That is also why we have not done a formal user study to test the effectiveness of the system from a user's perspective yet, but this is planned for the future.

The location ontology is used to be able to retrieve relevant resources according to the current user context. For this purpose we have designed a resource recommendation function that is utilizing the ontology graph to find other related resources such as persons or documents. In addition we have implemented a method to display current geo-references resources and points-of-interests on a map. The integration of POI types is also one of our extensions of the PIMO ontology.

Since manually entering address and location information is cumbersome on the mobile device, a semi-automatic solution is required. We have designed and implemented a solution to automatically learn relevant user locations based on the time-based clustering algorithm by [Kang et.at. 14]. Our evaluation showed that this approach works pretty well in our scenario and is capable to propose meaningful location instances for inclusion in the location ontology.

Future work includes the integration of more options to retrieve position and context in addition to RFID and GPS. One possible future enhancement is to use QR Codes for optical recognition where the user currently is located. Furthermore, an interesting idea is to include other user's personal information spaces in the retrieval process as well. People working in the same company may share common project and people in their personal ontologies. This information can be utilized to further improve the resource recommendation function [Woerndl et.al. 2009]. We are working on this extension of the SeMoDesk application towards the goal of a "social semantic desktop".

References

[1] Ashshbrook, D., Starner, T. (2002). Learning Significant Locations and Predicting User Movement with GPS. In *Proceedings of the 6th IEEE international Symposium on Wearable Computers*. IEEE Computer Society, Washington, DC, 2002.

[2] Chen, H., Perich, F., Joshi, A (2004). SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *Proceedings International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)*, Boston, MA, USA.

[3] Cheyer, A., Park, J., Giuli, R (2005). IRIS: Integrate. Relate. Infer. Share. In *Proceedings 1st Workshop on the Semantic Desktop*, ISWC Conference, Galway, Ireland, 2005.

[4] Ester, M., Kriegel, H.-P., Sander, J., Xu, X (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press. pp. 226–231, 1996.

[5] Kang, J.H., Welbourbne, W., Stewart, B., Borriello, G. (2005). Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9 (3) 58-68.

[6] Kay, J., Niu, W.T., Carmichael, D.J. (2007). ONCOR: Ontology- and Evidence-based Context Reasoner. *In: Proceedings 12th International Conference on Intelligent User Interfaces*, Hawaii, USA.

[7] Liao, Fox, L., Kautz, H (2007)..Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26 (1) 119-134.

[8] Marmasse, M., Schmandt, C (2000). Location-aware information delivery with comMotion. *In*: *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing,* Bristol, UK.

[9] Niu, W.T., Kay, J (2009). Location Conflict Resolution with an Ontology. In *Proceedings 6th International Conference on Pervasive Computing (Pervasive 2008),* LNCS 5013, Springer.

[10] Sauermann, L., Bernardi, A., Dengel, A (2005). Overview and Outlook on the Semantic Desktop. In *Proceedings 1st Workshop on The Semantic Desktop,* ISWC 2005 Conference, Galway, Ireland.

[11] Sauermann, L., Grimnes, G.A., Kiesel, M., Fluit, C., Maus, H., Heim, D., Nadeem, D., Horak, B., Dengel, A (2006). Semantic Desktop 2.0: The Gnowsis Experience. *In: Proceedings 5th International Semantic Web Conference,* LNCS 4273, Spinger.

[12] Sauermann, L., van Elst, L., Dengel, A (2007). PIMO – a Framework for Representing Personal Information Models. In *Proceedings of I-Semantics' 07*.

[13] Schulze, F (2009).*Context- and Location-aware Retrieval Methods for Personal Information Management on Mobile Devices.* Bachelor Thesis, Department for Informatics, Technische Universitaet Muenchen.

[14] Stahl, C., Heckmann, D (2004). Using Semantic Web Technology for Ubiquitous Hybrid Location Modeling. In *Proceedings First International Workshop on Ubiquitous GIS (TUbiGis 2004),* Gävle, Sweden.

[15] Woerndl, W., Groh, G., Hristov, A (2009). Individual and Social Recommendations for Mobile Semantic Personal Information Management, *International Journal on Advances in Internet Technology*, 2 (2&3).

[16] Woerndl, W., Hristov, A (2009). Recommending Resources in Mobile Personal Information Management. *In: Proceedings IARIA International Conference on Digital Society (ICDS 2009),* Cancun, Mexico.

[17] Woerndl, W., Schueller, C., Rottach, T (2007). Generating High Level Context from Sensor Data for Mobile Applications. *In*: *Proceedings IADIS International Conference Wireless Applications and Computing 2007,* Lisbon, Portugal.

[18] Woerndl, W., Woehrl, M (2008). *SeMoDesk:* Towards a Mobile Semantic Desktop. *In: Proceedings Personal Information Management (PIM) Workshop*, CHI 2008 Conference, Florence, Italy.