SeMoDesk: Towards a Mobile Semantic Desktop

Wolfgang Woerndl
Technische Universitaet Muenchen
Boltzmannstr. 3, D-85748 Garching
woerndl@in.tum.de

Maximilian Woehrl
Technische Universitaet Muenchen
Boltzmannstr. 3, D-85748 Garching
mwoehrl@gmx.de

ABSTRACT
In this paper we describe an application for personal information management (PIM) on a mobile device such as a PDA. While Semantic Desktop ideas to support PIM seem very useful in a mobile setting, most existing approaches are geared towards desktop computers. We present our SeMoDesk application from a user perspective and then explain some underlying concepts and details. Main features include a user interface optimized for PDAs, being able to automatically integrate resources such as phone calls and visualizing the information using tree, detailed or chronological views. A preliminary test showed that test users found the application useful.

Author Keywords
Mobile application, Semantic Desktop, ontology, mobile personal information management, resource management

ACM Classification Keywords
H.4.1 Information Systems Applications: Office Automation, H.3.4 Information Storage and Retrieval: Systems and Software

INTRODUCTION

In our information age, people use a lot of different applications, collect data such as documents, emails, bookmarks to websites and other resources, and utilize personal information management (PIM) tools. PIM is intended to support the activities people perform to organize their daily lives through the acquisition, maintenance, retrieval, and sharing of information [6]. One approach to deal with PIM is towards a "Semantic Desktop", aiming to integrate desktop applications and the data managed on desktop computers using semantic web technologies [4].

Organizing data and having access to relevant information is particularly important in a mobile scenario, e.g. field staff meeting customers. To support these tasks, PDAs (personal digital assistants) and other mobile devices are available. However, organizing information on mobile devices is even more difficult when compared to a desktop setting. This is mostly due to the fact that mobile devices have limitations in network bandwidth, storage capacities, displays and input capabilities.

Yet most of current PIM and Semantic Desktop research activities are not geared towards mobile information management. Therefore, the goal of this work is to adopt Semantic Desktop ideas for a solution on PDAs. The rest of the paper is organized as follows. First we discuss the background of our approach, including related work. In the following section, we present our SeMoDesk application from a user perspective. Then, we explain interesting aspects of the design and implementation. We finally conclude with a brief summary and outlook.

BACKGROUND

Semantic Desktop & Related Work

„Semantic Desktop" is an approach building on Semantic Web technology to improve the management of personal information (see http://www.semanticdesktop.org). The main idea is to assign meta data to all data objects that a user uses on her computer. Thereby, relations between resources can be defined with the goal to integrate desktop applications and enhance finding relevant information.

One Semantic Desktop implementation is the Gnowsis system [4]. 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.

Current research towards a "Social Semantic Desktop" is also carried out in the "Nepomuk" project (see http://nepomuk.semanticdesktop.org).

Other related work includes the Haystack project which aims at connecting application data and let people manage their information using personalization [2]. However the Haystack client is a rather complicated and extensive application that is not usable for mobile devices. "Stuff I've seen" is another interesting desktop application [1]. It allows creating an index of content, including Microsoft

Outlook resources, files, and Web pages in the browser cache.

On mobile devices such as PDAs, users usually manage their personal data using standard software such as Microsoft Pocket Outlook and other similar commercially available applications. We are not aware of an application that introduces Semantic Desktop ideas to mobile devices, in the manner described in this paper.

So while there is a lot of research going on in the areas of personal information management and the Semantic Desktop, most solutions are not carried out with regard to mobile devices, while the mobile scenario seems even more important for PIM.

Requirements for a Mobile Solution

Requirements for a PIM solution in the mobile domain include:

- A user interface tailored for mobile usage on a PDA
- The application has to be optimized for offline usage since network connectivity is not always available (or costly) in a mobile setting. Therefore, a mobile client connecting to a Gnowsis server, for example, may not be a practical solution
- Integration of phone and messaging (SMS, email) capabilities
- Due to the limitations of mobile devices (e.g. input capabilities), it is desired to automate tasks such as importing resources as much as possible
- „Re-searching“ [3] and retrieving relationships is important: for example, a field staff member has to be able to easily query past interaction and relevant documents when meeting a customer.

PIMO

The Semantic Desktop approaches rely on ontologies to formalize relationships between resources and define a concept hierarchy that can be utilized for information retrieval. For the Gnowsis project, the “Personal Information Model” (PIMO) ontology was designed [5]. We have based our application on the PIMO ontology, therefore it is explained in this subsection. 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 [5]. All objects in PIMO can be connected to each other using relations such as “hasPart”.

THE SEMODESK APPLICATION

In this section, we describe our application from a user perspective. SeMoDesk runs on Microsoft Windows Mobile 5 operating system which is common among PDAs. This type of device typically has a 240x320 pixel display with a touchsceen display and optionally a (small) keyboard. In addition, more and more PDAs are equipped with GSM and/or UMTS mobile phone capabilities.

Tree Structure of Concepts and Instances

After starting our program, the hierarchical concept ontology and assigned instances are displayed. The ontology is based on PIMO (see above), therefore it is also structured into “Thing” and “ResourceManifestation”. The concepts are denoted by customizable icons. The child nodes of a concept are loaded when the parent concept is expanded. By tapping on any concept, a menu is shown with options to display or edit the concept, delete it or adding subclasses and instances (Figure 1).

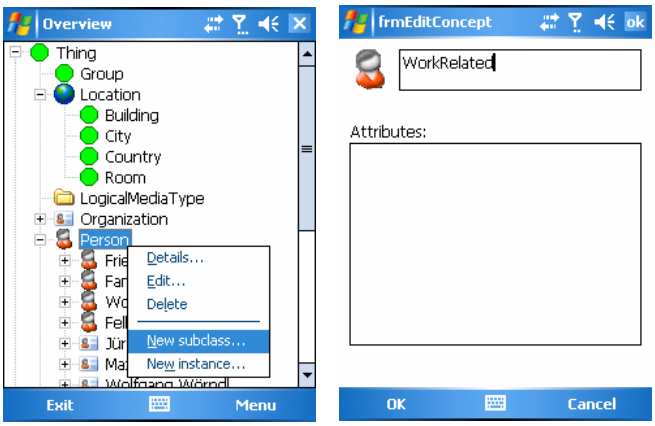

Figure 1. Creating a new concept as subclass of a “Person”

Data Sources

Resources in our program consist of all data types that are available on the mobile device. Resources are integrated by defining specialized data sources. We have implemented the following data sources: contacts, phone class, tasks, appointments, messages (including email and SMS), weblinks and a “file sniffer”. The file sniffer can be used to automatically detect new files on the device and import these files as resources in our application. Figure 2 shows how to configure the file sniffer to detect and import (new) pictures into the “ResourceManifestation” hierarchy.

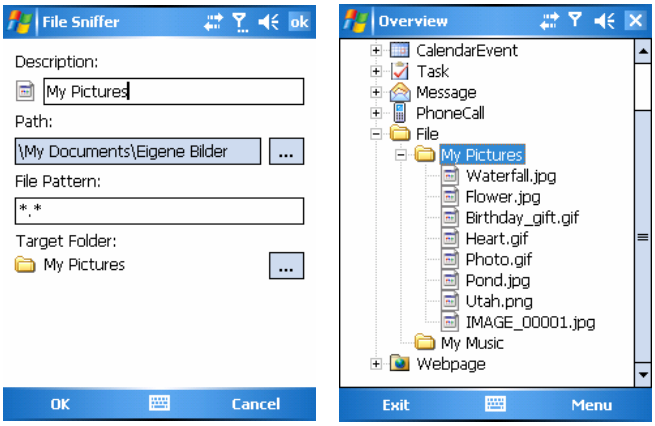

Figure 2. Configuring the file sniffer to import images

File sniffer sources can be added, managed and deleted by the user, all other data sources are fixed.

Figure 3 shows the user interface when managing an incoming phone call. Meta data such as date and time and the length of the call are integrated automatically. SeMoDesk also creates a relation with the caller if she is found in the address book resp. as instance of the "Person" concept.

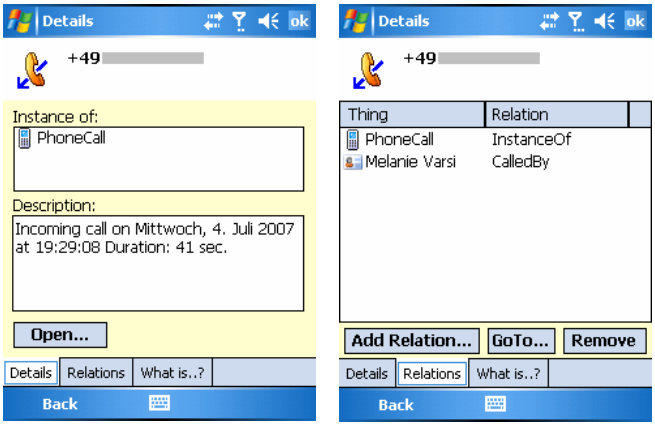

Figure 3. Details of resources, e.g. an incoming phone call

Relationship of Resources

In addition to automatically import resources, it is possible and desired to manually create relationships between resources, instances and concepts. Thereby, a user can assign a person to a subclass of "Person" (Figure 4, left), for instance. Other useful examples include assigning documents, persons, email messages etc. to "Projects", or assigning a room resources and presentation documents to a "CalendarEvent" (or vice versa) to easily retrieve room information and a presentation file to an appointment. Relationships can be defined in the user interface on any screen that shows the details of resources, e.g. Figure 3, right.

All generated and manually created relationships can be retrieved by displaying the details of an instance, for example all communication interaction with a person (and other relations) (Figure 4, right).

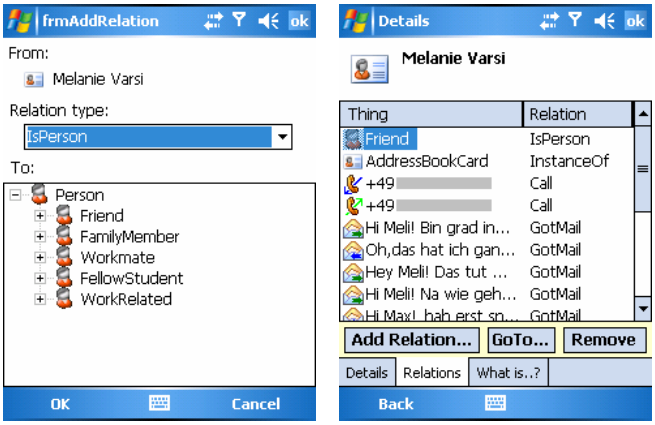

Figure 4. Creating a relationship and displaying existing relations for an instance

In addition to the structured view in Figure 4, it is also possible to display a chronological view of resources. This applies to all resources that more or less correspond to points in time, examples include: "CalendarEvents", phone calls, messages, and also files (modification date). The user can set the scale of the chronological view to either daily, monthly (shown in Figure 5) or yearly.

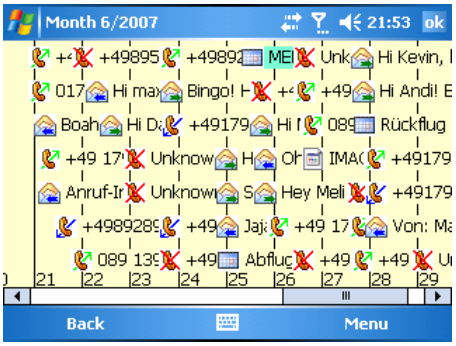

Figure 5. Chronological view of instances

DESIGN AND IMPLEMENTATION

After demonstrating our SeMoDesk application, we will now explain the underlying concepts and describe some aspects of the design and implementation.

Data Model

After informally introducing "resources" and other data entities in SeMoDesk, we now first define and explain the data model in our application:

- Resources: represent data on the mobile device, e.g. a file or calendar entry
- Concepts: abstractions of real things, organized in a hierarchical ontology. Every concept is directly or indirectly derived from the root concept „Thing"

- Instances of concepts are entities without a corresponding data entity (otherwise it is a resource), e.g. a concrete instance of a concept "My Project"
- Relations: connect concepts, instances and resources with each other
- Types of relations: describe what entities can be connected with what other entities, and what the inverse relationship type is

In our approach, we restrict the availability of relation types with regard to the involved data types, e.g. for a "Person" only certain types of relations are available, including "isPerson" (also see Figure 4, left). The main advantage is that the user interface becomes much clearer when only relation types are available that make sense in a certain context. The user has to choose only from a couple of relations because selecting from a larger list is cumbersome on a mobile device. In addition, the ontology is thereby kept clear of nonsensical statements. Our relations are also not symmetrical per se.

In general, relations are defined by stating a subject (e.g. an instance of a "Person" concept), the predicate (e.g. "isPerson") and the object (e.g. a sub concept of "Person" to structure the contacts), see Figure 4 for an example. This approach also follows the RDF (resource description framework) data model which has become a de facto standard for expressing meta data.

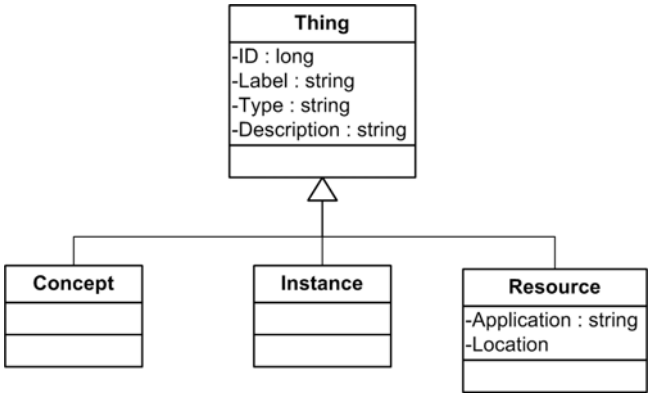

Figure 6. UML diagram of the class "Thing"

Class Design

While distinguishing between „concepts" and „resources", we treat them similar in our application. Therefore in our software design, there is a superclass „Thing"[1] and derived classes for concepts, instances and resources (Figure 6). Resources have additional attributes designating their location (e.g. path to file) and a corresponding application (see below in subsection "Data Sources").

Database Schema

For storing the data, we use a SQL database that is available for Windows Mobile. Figure 7 shows the database schema with the used tables and attributes. The table THINGS stores all resources, concepts and instances. The attribute "label" contains the text that is shown in the user interface, "IconIndex" is a path to an image that denotes this entity. In the table RESOURCES additional attributes of resources are stored, "location" is the actual path to the data item.

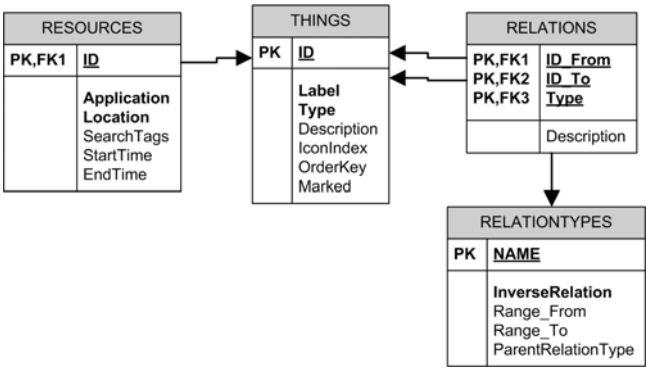

Figure 7. Database schema

Identification of Resources

One issue is the identification of resources. In the Web, usually URLs or, more generally, URIs (uniform resource identifier) are used. For our application, it is questionable how to address concepts that are not actual files, or other resources, but instances. Therefore, we do not use URIs to identify resources and concepts, but unique identifiers (UIDs). An UID is a number that is generated by using the current time and also a counter to allow for different IDs at the same point of time. One disadvantage of using UIDs instead of URLs is that it is necessary to store the "location" of a resource, in addition to managing the "ID" of things (see Figure 6, above), since the UID does not reveal the actual path to a resource like an URL.

Data Sources

As already mentioned in the previous section, every data source needs an implementation to import and manage resources. For these tasks, we designed a "clsSniffer" superclass and derived classes "clsFileSniffer", "clsMessageSniffer" and so on. The sniffer classes do include an UID attribute of the corresponding concept, e.g. the concept they are supposed to manage.

[1] Note the difference between the superclass "Thing" in the design of our application and the root concept "Thing" in the PIMO ontology

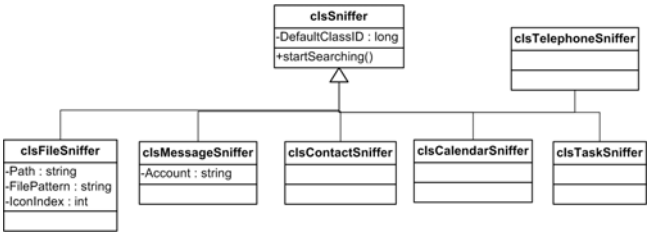

Figure 8. Class design for data sources

As an example the "clsContactSniffer" that manages the users' contacts is explained here in more detail, the other sniffer classes are realized analogous. Searching for new contacts is first done when starting the application. New contacts are marked in the personal ontology, so that the user can easily recognize them. When creating contact instances, available data items such as phone numbers and email addresses are imported automatically as well.

The "clsFileSniffer" could integrate additional meta data of data sources. For example, JPEG pictures may contain EXIF (exchangeable image file format) meta data, with information on the date and time of the picture, camera model etc. that could be imported and utilized.

Implementation

The application was developed using the Microsoft Visual Studio 2005 IDE for .NET Compact Framework 2.0. The used programming language was C#, the underlying database is Microsoft SQL Mobile 2005. SeMoDesk runs on every Windows Mobile 5 device with installed .NET Compact Framework and SQL Server 2005 Mobile components.

The access to Microsoft Pocket Outlook resources is done via the "Microsoft.WindowsMobile.PocketOutlook" namespace. However, this interface allows only access to the address book, tasks and the calendar in Outlook. To automatically integrate phone calls and (email) messaging, an additional external component is needed. We thereby used the commercial library "Mobile in the Hand" (see http://inthehand.com/content/Products.aspx). Since this product is not free, we did integrate this library in a separate module. Therefore, the main SeMoDesk program can be used without a "Mobile in the Hand" license, but also without integrated import of phone calls and messages.

Evaluation

The application has not been formally evaluated in a user study so far. However, we conducted real life tests with several test users, the screenshots in this paper are from actual usage. We tested the SeMoDesk application according to the following factors:

- Correctness and stability: does the system work as advertised, or are there any errors or crashes?
- Performance: does the application run fast enough on a mobile device (PDA)?
- Usability of the user interface
- Usefulness: is the application useful, does it provide beneficial value in relation to the required effort? Which features make sense?

In the first version of the application, there were a few bugs and crashes. After fixing the errors, SeMoDesk ran very smoothly and provides the functionality described in this paper, without any major problems. The performance of the program is good, starting up takes about 10 seconds but afterwards the applications runs pretty fast without much delay when importing resources, for example. We did not notice any decline in performance with more and more data added to the application.

The usability of SeMoDesk was rated well by test users. What was missed was drag-and-drop functionality people are used from desktop applications, but are difficult to provide on a mobile device without a computer mouse. One minor issue that is not possible at this time to define relationships with a number of resources in one step. This has to be done for every resource individually which can be a bit laborious. Overall, the test users rated the application as very useful. They liked the idea of structuring data according to an ontology as it eases re-finding information. The automatic creation of relationships of resources such phone calls to contacts, for instance, was especially rated very highly. On a mobile device, it is beneficial to disburden the user from as many manual tasks in the user interface as possible.

CONCLUSION

The main motivation for this work was that while Semantic Desktop ideas seem very useful in a mobile setting, most existing approaches are geared towards desktop PC usage. Therefore our goal was to design a useful mobile application that allows user to define, manage and utilize a personal information ontology on her mobile device, in our case a Windows Mobile PDA. We have presented the user interface and explained the design and implementation of our SeMoDesk application. One of the main aspects was to make the application as usable on the mobile device as possible, which includes automatically importing resources.

Future prospects include the design and implementation of a corresponding desktop application including synchronization, and/or integrating with other Semantic Desktop approaches. One reasonable scenario is that the user defines her PIM ontology on the desktop and imports and manages resources on the mobile device, while being able to occasionally add sub concepts on the PDA as well. Some resources may be shared between desktop and mobile

computers, and others may be only available in one environment. For the latter case, we think of "virtual resources", i.e. resources which are an instance of the hierarchy, but not currently available.

Other planned future work is towards improving the retrieval of information. We think of functionality to analyze and interpret the data, e.g. intelligent searching. The goal is that the system automatically detects resources that may be of interest to the user in a given context. We are thereby working on context-aware recommender systems [7] and plan to integrate some ideas in this project. For example, the system could detect that a user is in a meeting, according to her calendar and the current location and time. If the meeting calendar entry is associated with a project X, then the application could offer quick access to documents (or other objects) that are also instances of project X.

The visualization of concepts and resources could be improved. For example, the chronological view (see Figure 5) has to be more clearly arranged and better adapted to the actual data that is visualized, to make it more useful. Analogous to chronological views, spatial information could be shown in a map, e.g. the position of rooms, or any other data entity that can be assigned to a locality.

REFERENCES

1. Dumais, S.T., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R., and Robbins, D.C. Stuff I've Seen: A system for personal information retrieval and re-use. *Proceedings of SIGIR*, 2003
2. Karger, D.R., Bakshi, K., Huynh, D., Quan, D., and Sinha, V. Haystack: A General-Purpose Information Management Tool for End Users Based on Semistructured Data. *Proc. Conference on Innovative Data Systems Research (CIDR)*, Asilomar, CA, USA 2005, pp. 13-26
3. Marshall, C.C., and Jones, W. Keeping Encountered Information. *Communications of the ACM*, Vol. 49, No. 1, 2006, pp. 66-67
4. Sauermann, L., Grimnes, G.A., Kiesel, M., Fluit, C., Maus, H., Heim, D., Nadeem, D., Horak, B., and Dengel, A. Semantic Desktop 2.0: The Gnowsis Experience. *Proc. 5th International Semantic Web Conference*, Springer LNCS 4273, 2006
5. Sauermann, L., van Elst, L., and Dengel, A. PIMO – a Framework for Representing Personal Information Models. *Proceedings of I-Semantics' 07*, 2007
6. Teevan, J., Jones, W., and Bederson, B.B. Personal Information Management. *Communications of the ACM*, Vol. 49, No. 1, 2006, pp. 40-43
7. Woerndl, W., and Groh, G. Utilizing Physical and Social Context to Improve Recommender Systems. *Proc. Workshop on Web Personalization and Recommender Systems (WPRS), Web Intelligence conference*, Silicon Valley, USA, Nov. 2007

The columns on the last page should be of approximately equal length.