Evaluating Long-Term Use of the Gnowsis Semantic Desktop for PIM

Leo Sauermann and Dominik Heim

Knowledge Management Department
German Research Center for Artificial Intelligence (DFKI) GmbH,
Kaiserslautern, Germany
`{firstname.lastname}@dfki.de`

Abstract. The Semantic Desktop is a means to support users in Personal Information Management (PIM). Using the open source software prototype Gnowsis, we evaluated the approach in a two month case study in 2006 with eight participants. Two participants continued using the prototype and were interviewed after two years in 2008 to show their long-term usage patterns. This allows us to analyse how the system was used for PIM. Contextual interviews gave insights on behaviour, while questionnaires and event logging did not. We discovered that in the personal environment, simple has-Part and is-related relations are sufficient for users to file and re-find information, and that the personal semantic wiki was used creatively to note information.

1 Introduction

People gather information on their desktop computers, but current systems lack the ability to integrate this information based on concepts or across applications. The vision of the *Semantic Desktop* [16] is to use Semantic Web technology on the desktop to support *Personal Information Management* (PIM). In addition to providing an interface for managing your personal data it also provides interfaces for other applications to access this, acting as a central hub for semantic information on the desktop. *"Here I have the possibility to gather things"* is a quote from a user of our prototype. Previous work published about Semantic Desktop applications [3,15,17] did show that this approach is promising to support users in filing and finding information, and to work with information in new ways. The *challenge* of our field is that evaluations with real end-users are scarce, and especially there exist no long-term study of Semantic Desktop usage. In this paper, we present two long-term evaluations of the *gnowsis* system.

The rest of the paper is structured as follows: First, the underpinning ideas of the Semantic Desktop and the *gnowsis* implementation are described. The *research question* is to see if our *PIMO ontology* and the *software prototype* were used by the participants for PIM tasks, namely filing and finding information. A short introduction to the methods of HCI evaluations and the problems faced when evaluating PIM system is given, which influenced our selection of tools. In Section 2 a two-month case study with eight participants is presented. The

second evaluation is a contextual inquiry after 22 months with two users who continued using the system, in Section 3. A discussion follows in Section 4. The paper concludes with an outlook on future work. Readers should be familiar with the approach of the Semantic Desktop which is described in the related work.

1.1   Semantic Desktop and Gnowsis

In [17] we presented the basic architecture of the evaluated system. The core services are: a service to *crawl data* on the desktop and convert it to RDF (the Aperture[1] framework), store data in an *RDF database*, and *infer new knowledge* from the data. A *search* service provides fulltext and semantic search, a *semantic wiki*[2] provides means to store text, and an *ontology service* provides a programmatic API to *tag* documents or manipulate classes and instances. These services have been described in [17] and were further improved and extended in the NEPOMUK project [7].

The underlying ontologies consist of generic ontologies for files, e-mails, and other document types, and higher level ontologies to represent the mental model of the user, our application domain. The *Personal Information Model* (*PIMO*) [18][3] is a model to represent a single users' concepts, such as <u>projects</u>, <u>tasks</u>, <u>contacts</u>, <u>organizations</u>, allowing files, e-mails, and other resources of interest to the user to be categorized, independent of application and with multiple relations [4]. It is based on RDF/S, it was used as main information representation ontology for the end-user. The generic upper-class is <u>Thing</u> and we will refer to it throughout this paper. In Section 3.1 the predefined sub-classes of <u>Thing</u> are listed. They were selected based on their generality and applicability independent of domain. Latif and Tjoa [12] came to a similar selection that influenced us.

In the rest of this paper RDFS classes, instances, and properties defined in a user's PIMO are distinguished from text by <u>underlining</u>. It is a reference to hyperlinks and emphasizes the fact that each element points to an RDF resource and can be browsed by users.

Based on the core services and the PIMO ontologies, various applications exist to help the user filing and finding information. A short description of each component is given now. *Miniquire*[4] is a sidebar application showing an overview of the user's PIMO. As shown in Figure 1 the sidebar contains a search interface on the top and below an overview on the user's PIMO. It allows users to quickly find things inside their PIMO or manipulate them. It contains functionality for adding and deleting (sub-) classes or things as well as more sophisticated options like hiding or highlighting things.

The *Thing Editor* shown in figure 2 enables the user to focus on a specific <u>Thing</u> to see all relations as well as providing the possibility to edit the relations

---

[1] `http://aperture.sourceforge.net`

[2] The open source Kaukolu wiki: `http://kaukoluwiki.opendfki.de`

[3] An improved an more current draft version is available at
`http://dev.nepomuk.semanticdesktop.org/wiki/PimoOntology`

[4] The name is a pun referencing the work that was before, Tim Berners-Lee's Enquire.
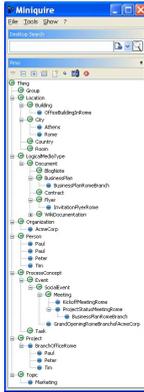
Fig. 1. The sidebar user interface "Miniquire"
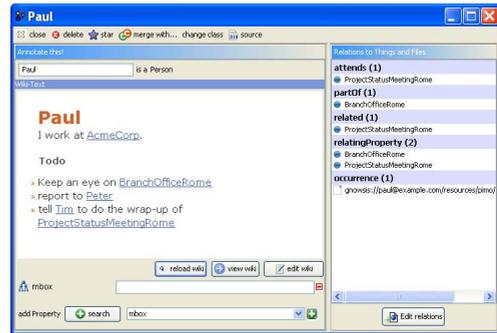
Fig. 2. The "ThingEditor" browser and editor

and meta-data. To the left the semantic wiki *kaukolu* is embedded. On the right side, the relations are shown as a list. Clicking on the related things navigates to them. When editing, the core relations easily editable are (defined in the PIMO ontology): <u>related</u>, <u>part Of</u>, <u>has Part</u>, <u>has Topic</u>, <u>is Topic Of</u>. More properties can be added. The *Personal Semantic Wiki* is realized using the *Kaukolu* [11] open source software. The main idea is that a wiki page can be created for every <u>Thing</u> in the user's PIMO. Using a semantic wiki syntax, it is possible to annotate concepts.

As identified by Barreau and Nardi [1], support for filing information is a crucial task in personal information management. The *Drop-Box*[5] is an application to (semi-) automatically move and classify a file. It consists of a normal file folder called "dropbox" that is observed by the gnowsis system. It usually resides on the desktop. If a file is dropped, a "Drop-Box" window appears showing possible tags to classify the file and possible folders where to move it. New tags and folders can be created ad hoc. The system *suggests possible tags* based on text analysis and document similarity to previously tagged documents[6]. Two *Plugins* were developed for Mozilla Thunderbird and Microsoft Outlook to tag items in these applications. These plugins don't support recommendations.

At this point it is important to know that all interfaces work on the same level of abstraction — the <u>Thing</u>. The name of a *wiki page* equals the name of the <u>Thing</u> equals the name of a *tag*, and can relate (if assigned) to a file folder on disk. This makes a user's PIMO a data structure used across applications and data domains. Another application is a *Semantic Search* interface combining full-text search and semantic search. It was described in [17], the latest version is [19]. More details about the software and its features can be found in [17], or by watching the tutorial

---

[5] The name is derived from the Mac OS drop-box.

[6] For a detailed description, refer to our previous publications or `http://www.gnowsis.org/statisch/0.9/doc/use_cases/_dropbox.html`

videos[7]. The software used for the evaluations is *gnowsis* 0.9[8]. It is written in Java and is available under the open source BSD license.

1.2   Related Evaluations

There are different approaches to evaluate software in Human Computer Interaction (HCI). One approach is to set up an experiment in a lab-environment, reducing factors that may influence the experiment and invite test users to use the system in a supervised way. Such evaluation is typically done in a short time span (one day, or a few hours).

In a well-noted paper, Greenberg and Buxton [6] questioned if this practice of HCI evaluations is good for all cases. Their key argument is *the choice of evaluation methodology—if any—must arise from and be appropriate for the actual problem or research question under consideration. Eat-your-own-dogfood* is one of these possible methods.

In the field of PIM, the main problem is the long-term nature of any activity. Storing and retrieving tasks cannot be observed in laboratory settings in a realistic way. For PIM, privacy concerns are a problem and also the stability of the software prototypes, as people will depend on the usefulness of the system when evaluating it. Kelly [10] showed that in PIM, an evaluation has a longer duration and aims at providing information about the usefulness[9] of the software itself, and not only the usability of the GUI. She also recommends not to evaluate with fictional tasks, as PIM is highly associated to real user activities. Furthermore Kelly showed that it is important to let the user work his or her own personal space of information. In related work on PIM [15,2] we find that evaluations based on interviews with users, accompanied by implementing a prototype are common.

In terms of quality, Nielsen [14] argues that five users are sufficient to discover 85% of the usability problems in an interface. Many insights about Microsoft's "MyLifeBits" [5] prototype were found with one very dedicated user, Gordon Bell, who is also co-author of the publications.

1.3   Decisions for the Gnowsis Evaluation

We agree with the view of Greenberg and Buxton and consider *eating your own dogfood* and long-term contextual inquiries as methods that help us to learn about Semantic Web technologies. Our research question is to see how our *PIMO ontology* and the *software prototype* were used by the participants for PIM tasks, namely filing and finding information[10]. Especially it is important to know how users file information, find and re-find information, and maintain their structures using the Semantic Desktop. Also the relation between mental models and the explicit PIMO structures is interesting. Secondary it is interesting to get feedback about the usability of the software.

---

[7] `http://gnowsis.opendfki.de/wiki/GnowsisUsage`

[8] `http://www.gnowsis.org/Download/`

[9] Distinguishing between usability and usefulness is stressed by [6].

[10] These tasks are considered essential PIM activities in [9].

2   Usability Evalution July 2006

To find the usage patterns of the users, a two-month evaluation was planned with participants using the software on their own computers in their daily work.

2.1   Participants

In the beginning of the evaluation, we asked the participants at a scientific workshop to participate in the experiment, 15 agreed. When the actual evaluation period started, only a small fraction of the initial volunteers was still available. To get useful results, 8 volunteers on site where then asked to participate. Compared to remote users, these participants were available for the contextual inquiry interviews at the end and for the usability test at the beginning. Also, it was possible to fix software problems in-situ. More background information about the selection of participants is given in [8, p81]. The participants were not financially compensated for their effort.

All of the participants worked within DFKI, our company, two were from departments that are not related to Semantic Web, six were related to Semantic Web and our Knowledge Management Lab. This biased them to rate the prototype better than it actually was, but also let them be more forgiving when bugs and problems occurred. Their ages ranged from 25 to 40, one participant was female. All participants were familiar with desktop computers and general PIM activities. All participants but one were native German speakers, their feedback was translated to English by the authors of this paper.

Participant A was a male senior researcher who was also occupied with software project management and consulting. He installed gnowsis in early 2006 and was still using it in May 2008. He has experience in Semantic Web technologies and semantic modelling. He was using *gnowsis* often each day. His work documents include 8300 files within 1160 folders. He created 1196 elements in his PIMO. B was a female junior researcher engaged in writing a PhD thesis and research project work. She has been using *gnowsis* since July 2006 and was still using it at the time of writing. She has experience in Semantic Web technologies and semantic modelling. Her work documents include 75900 files and 11700 folders. She created 465 elements in her PIMO. Participant C and D were student workers of other departments who had programming experience but did not know about Semantic Web. The four remaining participants were researchers from the Knowledge Management department.

2.2   Procedure

The evaluation started with the installation of the software on the user's desktop in July 2006. Then first interactions with the user interface were explained by the interviewer and done by the user. We used a software tutorial to guide users through the first steps with the system and make them acquainted with the features, which we called *use cases*. This was done for each participant individually at their office. The interviewer took written notes of the user's feedback

and encouraged users to *think aloud*. This proper user training took about one hour for each participant, but turned out to be an important and time-saving step considering the following two month long-term evaluation.

The *team approach* introduced by Morse [13] is an idea to accompany an evaluation with sessions during which the users exchange their experiences with the software. Users explained each other how to use the software and shared their experiences, how to solve practical problems with it, which turned out to be a good support of the users.

In parallel, an *activity logger* collected statistical data of which actions the user did with the system. The last part of the evaluation was a *final contextual inquiry*. Contextual Inquiry (also known as proactive field studies) is a structured field interviewing method that aims to fully understand the users working-environment. The evaluator can ask each user individual questions to collect information about the processes themselves as well as the consequence of the behaviour [10].

In former evaluations (in 2005 and January 2006) we realized the importance of direct conversation with the participants. The interviews were conducted in the office of the participant, while the user was working with the system. The inquiry consisted of two parts. The first one mirrored the *training use cases*, asking when and how the participant used each feature. The user had to rate each use case by importance and frequency of usage as well as the most often used features. In the second part we asked for information about the frequency of using the software as well as features the participant missed during his interaction.

One of the most important question to us was for what tasks and goals the software was used, and how they relate to PIM. Given such a generic tool as the Semantic Desktop, what problems will users solve with it, and what creative ways did they invent to reach their goals?

2.3   Results of the Usability Evaluation 2006

The results of this evaluation can be divided into three main parts: *Expectation Questionnaire*, *Usability and GUI*, and *PIM use cases*. The *activity-log* file was used to cross-validate answers.

The *Expectation Questionnaire* consisted of ten questions that were answered using a six-point *Likert-scale* about expectations and ten yes-no questions about previous experience. Users emphasized that the system should be easy to use, help structure their documents, and provide a good search function. Interestingly, more than half of the participants used desktop search engines and nearly all have used wikis before. The rest of the results can be found in [8, pp85-87].

Results from the *Usability and GUI* inquiry were divided into positive and negative feedback. The spoken feedback from the contextual inquiry at the end and during the evaluation was gathered, categorized and grouped. This results in a "most mentioned features" list. The autocomplete functionality of the wiki was mentioned first. It suggests what things can be inserted in the wiki after the user has typed the first letters and pressed a key combination. Second was

miniquire and Thing Editor.This was followed by the drag-drop functionalities, "starring" things, and the easy installer.

Negative feedback was about the slow search and the inability to stop a search once started. The need to switch from the browser-based wiki to the Java-Swing based Thing Editor was described as a "loss of focus". Users wanted to filter the ontology tree in miniquire. This was added later and turned out to be a key feature for the two long-term users. An in-application help-system (tooltips, help buttons, manual) was also requested and found missing. All users noted problems with parts of the GUI.

Results from the *activity logger* showed that one user had the system running permanently, one user 3 hours per week, two user 2 hours, one user 1 hour, and one user 30 minutes per week. The rest of the activity log was used to see what features the participants used, but did not help us much to find out if the system supports PIM or not. The *Semantic Wiki* was used by 75% of the participants for note-taking and one third noted that semantic relations can be created faster using the semantic wiki syntax than using the graphical Thing Editor. Half of the participants did not use the *e-mail tagging plugin*, because either they use another email client or no local client at all. 25% of the participants used the plugin but stopped because they did not get any advantage of it (e.g. retrieval of all tagged emails is not possible).

The *most frequently used* components were (multiple denomination was allowed): 87.5% used Miniquire, 75% the wiki, 25% the Thing editor, 12.5% the DropBox. This was also backed in the activity log-files. The rest of the results of the first contextual interview is presented in [8, pp87-93]. About the usefulness of the features, participants were very positive, but cross-validation with the activity log showed that some participants did not use the rated feature more than 10 times and therefore these results are not relevant.

The results of the evaluation logger show, that users did *extend the default PIMO* ontology with custom classes, 11 classes in the mean with a mean derivation of 9. Thus, some users did create many classes while other none. Only half of the users created custom properties, and then not in a significant amount (less than five). Users did create instances though, altogether 371 instances, but with a mean derivation of 81 between users, so some were not active at all.

Main Purpose Of Usage. The purpose of usage divides users. 37.5% of the participants stated to use gnowsis for project management, but 25% also stated explicitly *not* to use gnowsis for project managemant, as it did not provide sufficient features for them. The majority of the participants used *gnowsis* primarily for managing purposes (e.g. events, conferences). This is not surprising, as *gnowsis* is a tool for the management of personal information. This is approved by 37.5% of the users, who used gnowsis to organize their knowledge (including their ideas). The fact that relations within PIMO can help while creating new knowledge is documented by 12.5% of the participants that used gnowsis as support during writing papers.

2.4 Discussion of the First Evaluation

A discussion of the results of this evaluation is given now, more results from the second evaluation are shown later. The most feedback was given about the wiki, miniquire, and the Thing Editor, so we focus on them.

Semantic Wiki. The expectation questionnaire already showed that the majority contribute to wikis as well as use them to organize themselves. This was also reflected during the evaluation. The majority of the participants used the wiki as personal notepad. Some of them used it for entering text as a notepad and some for entering text in a quick manner that was remodelled as PIMO structures later on. The auto-completion feature helped to add relations with a minimal effort. The participants found manifold fields of application for the wiki: documentation, comments on files, contact information of persons, to-do lists, notepad.

Miniquire and Thing Editor. When browsing and annotating in the Thing Editor, the basic default relations of PIMO seemed to be enough for the users to structure their knowledge in a way that allows them to retrieve it. Here the simple graphical view of related items in a list to the right did help users to navigate. Users did not create custom relations, but did create custom classes.

The question is now, how users fulfil their PIM tasks using the system. Based on the two months of use and the questionnaires, we can say that the system assists in PIM, but not how exactly and what the key semantic features are.

3   Contextual Interviews April 2008

Nearly two years after the first evaluation, three users were still available and kept using the system. They were interviewed in a contextual inquiry in April 2008. One of them did only sporadically use gnowsis for daily work and is excluded from the results. The participants *A* and *B* (described above in 2.1) remain, to keep their anonymity we further only speak of "one participant" in the male form. Both participants described themselves humorously as *nitpicking information keepers*, one used the German term "*Strukturierungszwang*". Also, both are colleagues of the main author of this paper. They should be seen as "*eat your own dogfood*" users, that are biased positively towards the system and behave enthusiastic about it. Other users do not share this enthusiasm, only 2 of 8 have continued using the system voluntarily. Nevertheless, the behaviour the users show and the structures they created in their PIMO tell us something about the use of ontologies for PIM.

As *procedure* we concentrated on a contextual inquiry, as this method brought the most interesting results in the last study. Instead of taking written notes we used a video of the interview. The contextual interviews started with setting up a video-camera for audio and screen recording and an introduction to the process. Basic questions about the participant were asked for warm-up. These were questions about name, gender, occupation, since when and how often they

use gnowsis. Going from the warm-up into the contextual part was by the question *For what did you use gnowsis last?*. Then participants began showing their structures and telling about experiences. If needed, specific questions were asked about the PIMO classes and instances they created, whether they associated files with instances, their web bookmark keeping behavior, usage of the wiki, and how they created associations and instances. Then, the interviewer asked participants to continue working on a task that they need to do at that moment anyway as part of their normal work. At the end of the inquiry, feedback about the interview process could be given and the interviewer asked the participants to provide a copy of the *Evaluation Logger* logfile that contains the activities the users have been doing. Altogether, each interview lasted for about two hours including a coffee break.

3.1   Results

The interviews were screened and transcribed to text, some open questions were asked two weeks after the interview to clarify facts. In the following observed usage patterns are presented and classified into filing, finding, and maintenance activities. A similar classification was used by [1] and is proposed in [9]. First observations about the PIMO structures created by the users are given.

Classes and Instances. The predefined classes of PIMO are [18]: Group, Location, Building, City, Country, Room, Document, BlogNote, Contract, Organization, Person, Event, SocialEvent, Meeting, Task, Project, Topic.

Users extended them with the following classes: Application, Domain, Hardware, Book, Notes, Paper Collection, Presentation, Proceedings, Project Documents, Project Plan, Project Proposal, Survey, Paper, Story (war stories, usage stories), Tips Tricks, Diploma Thesis, Thesis, PhD Thesis (or Dissertation), Project Work (a document), Pro-Seminar, Department, Research Institute, External Project, DFKI Project, Private Project, Conference (an Event), Conference Series, Phone Call, Workshop, Work Package, Student.

There may be more custom-created classes (but they were not shown to us). In the activity log we find that one user created 10 classes, the other 31. The question is, why and when users create classes. One user said about the possible subclass of Organisation, Ministry: *For example, I could have created "Ministry". But the effort to create it without having a benefit for it was not worth it. . . . I model when I think that I can use it. Like Research Company. When I had 2-3 research institutes amongst the companies the pressure was big enough — I created research institute and changed the class (of the existing instances).* The same case was with External Project. The participant started with the predefined Project assuming the semantics of "my own project" until faced with projects run by others. Then, another class was created for those. Both participants did not use the system for *private data*, although one created a class Private Project but did not create instances.

The class Location was scarcely used. Participants articulated no need for locations as they do not classify information by location. Upon further inquiry

Table 1. Interpretation of relations

Domain	Predicate	Range	Interpretation
project	has part	person	Person is member of the project.
organization	has part	person	Person works for organization.
project	has part	workpackage	The workpackage is part of the project plan.
meeting	has part	person	Person attends the meeting.
topic	has part	document	Topic isTopicOf document.
project	has topic	topic	Project is about this topic.

they both have used the Google-Maps integration of locations at the beginning, but the lack of support to automatically geo-code and create locations made it unattractive.

Also the possibility to create explicit Groups to collect similar things was not used, instead participants used to collect things by making them part Of another thing. One participant created 43 Meetings, the other 2, with the explanation *if the Outlook integration worked better, I would use it more.* Both participants used automated features of gnowsis to create Persons from address book entries. But at a later stage, this feature turned out to be buggy and was not used but maintained persons by hand. One participant entered telephone numbers, and e-mail addresses into the system. 101 and 154 persons were created. Both participants used Topics to classify things (59 and 201 instances). Especially people and documents were annotated with topics.

In total the first participant created 288 instances with 92 of them having wiki-pages, the other 959 with 148 wiki-pages. Most instances were created in the classes Topic, Person (and subclasses of person), Document, and Organization.

The system provides *basic relations* which were used extensively: part-of (used 566 and 63 times) with the inverse is-part-of, related (122, 193) has-topic (401, 78), and the inverse is-topic-of. Besides these generic relations, more precise relations were available but the user interface supports the basic relations better. Hence the basic relations were used more and interpreted depending on context, see Table 1. Over the longer period of two years, these structures are getting blurred and unprecise. Although the participant had no problem finding elements and navigating, he noticed that the structures are "wrong". An example was a final meeting $FinalMeeting$ about a report $Report$ for company $Acme$. A kickoff meeting $Kickoff$ started the process. The structures were (in simplified N3):

```
Kickoff a Meeting;
  hasPart PersonA, PersonB, PersonC, PersonD, FinalMeeting;
  partOf  Report;
  related TopicA, TopicB, ReportX;
  occurrence fileA, another_report_about_topicA, interviewnotes.
FinalMeeting a Meeting;
  hasTopic Report;
  partOf Report.
```

When looking at these structures during the inquiry, the participant noted that the has-topic relation between $FinalMeeting$ and $Report$ is redundant and

removed it. Later the participant said he created the relations over time, to navigate faster.

Filing Information. Following is behaviour specific to filing information, in the tasks the participants have chosen to work on during the inquiry. One participant chose to create a Person representing a new student worker. He created an instance of <u>Student</u>, added firstname, lastname, and fullname ("for the search"). A relation to a project was created and a folder on the hard-disk associated. The participant wished to enter the skype-id of the student, but was not able as the property was missing, and the user interface made creating properties complicated.

Another task was to create a new task for this student. An instance of <u>Task</u> was created, then opened and the previously created student selected as <u>related</u>. To express that the task is about a certain topic, two topics were assigned.

Participants used their filesystem and e-mail system in parallel with gnowsis. As e-mail was not integrated well (the plugins had many bugs and were de-activated by the users), participants did not annotate e-mails, but expressed the dire need for annotating e-mails. One participant used *gnowsis*' web-tagging tool to file websites. With files, the drop-box application saves time and helps decision-making when filing, therefore it was used much. The quality of automated tag suggestions was described as very bad, participants complained that they often had to do the tag assignment by hand. The rate of files annotated with gnowsis varies from folder to folder and application. For example, one folder with scientific papers was annotated heavily, whereas others weren't. Compared to file keeping using folders and folder hierarchies, the possibility of multiple classification was both heavily used and expressed as very positive both for filing and for retrieval. Both used the Drop-Box extensively, 386 and 149 times.

Noting text in the *personal semantic wiki* proved to be a key feature. Each thing in the ontology can have a correlated wiki page. Participants used the wiki page for different purposes: to write short notes defining what the concept is, longer notes with copy-pasted quotation and text snippets or to write down meeting notes. The wiki was also used to create web link collections by copy-pasting URIs into the wiki text (by the user who did not use the *gnowsis* web-tagging tool). Both participants discovered many hidden features of the Semantic Wiki by reading the documentation and used them creatively.

One participant created an instance of <u>Task</u> called <u>Todo</u>. On its wiki-page, he used the option of Kaukoluwiki to include other pages, and sections of other pages. The participant then created todo-sections in various other pages and included them all in the master <u>Todo</u> page. For example, the section *Things to do* in the wiki page of <u>ProjectX</u> was integrated in the master todo page. The inclusion was never removed or maintained from the master page, included pages do not show up once they are empty. This system allowed the user to *gather all todos in one place.*

Finding and Reminding. Both participants used the miniquire visualization as main entry point to the system. It is possible to filter the tree using

text-search. This feature keeps the spatial arrangement of information of miniqure but filters it. At the end of the inquiry, one participant noted that the miniquire search box is most important for him.

During the interviews, both participants did always use miniquire when opening a specific thing. Once a thing was open, the linked relations to other things were used to navigate. One participant described a certain PersonX as *entry point* to more data. It was an external project partner being responsible for a certain part of the project. The participant navigated from PersonX to find telephone notes, workpackages, and documents related to the project. It seems that once a certain path to information elements is followed, it gets trained and revisited many times later. The preference to follow paths and navigate amongst items was already noticed by Barreau and Nardi [1], "*users prefer to be able to go to the correct location on the first try*". Using fulltext search was often used when navigation fails.

One participant mentioned to gather information about a person X before doing an important business phone call. The relations allowed to step from X to previously entered phone notes, and to the project. When asked about *what would you do if gnowsis would be taken away from you*, the participant said that the missing text-search functionality would not be such a problem, but the structures and relations. He would not have *confidence* in himself when searching files, because he relies on the structures.

Maintenance activities. In PIM research, maintenance activities are tasks to reorganize or think about information [9]. One effective example use of the system for maintenance was the preparation for a survey. The participant did come back from an interview, having taken notes as a text. Later, he had to deconstruct the interview into parts. This is described as a "creative step": to read the transcribed text and create relations from the interview to other things, such as the project or the customer. For example, the topic S was brought up in a meeting by a customer. In the notes, the topic was mentioned but the participant did not know about it. As the customer will likely mention it again, he decided to create S as a Topic and add some text to it. An internet search brought up the homepage of organization SU which works primarily in S. The participant changed the type of S to Company and attached the homepage. Two documents describing S were related. The participant mentioned that S will be important for other projects also, that was the main reason to create it in his PIMO as an instance. If not for the possibility to relate it to future projects, he *would have just added some text about S in the wiki.*

Generally, participants needed some time to learn how to model effectively. One participant decided to start a completely new PIMO after four months of use, to clean up.

From the larger pool of available features, the two long-term users both only used a very limited set. We learn that these features help them to fulfill their day-to-day PIM. Namely, the miniquire view as an entry point, the wiki to keep notes (and search them), the relations between things, and the relations between things and files and folders are used.

3.2   Perceived Structures Versus Real Structures

Both participants knew their PIMO very well. They verbally used the terms as written in the PIMO. Upon asked what a certain thing in the PIMO represents, participants gave verbal explanations that are very similar to the wiki texts they have written. This confirms the nature of "supplement to memory" we envisioned in the first definition of the Semantic Desktop [16]. For both final participants, it was apparent that they can use the PIMO structures effectively, without always being modelled precise, correct, or exhaustive. They were able to find things effectively using the miniquire sidebar. But their *perceived mental model* of the PIMO structures differed from the *stored ontology*. Especially the relations written in PIMO are different from the mental models.

One participant said: *If I have a topic, I always know whom to ask.* Upon inquiry if this was in the past used to contact people, the participant did first only remember the first name of a person and showed a wrong person, and later remembered the right success story. Another example: *This person, Donald X, when I was looking at his homepage, I added the topics he works on.* Looking into the data, the topics were not annotated but the papers published by Donald X.

4   Discussion

For PIM we can conclude that the combination of wiki, tagging system, and ontology is a good basis to the Semantic Desktop. The wiki was the second most used component and all of the users did use it during the evaluation.

Looking at the small number of custom-created classes and the even smaller amount of created properties shows that the granularity of semantic modeling is not so important when used for **personal** information management. Users did remember where things can be found and how to navigate to them, and followed paths along "entry point" things. For the navigation to work, the nature of the relation (part-of, is-related, has-topic) is not relevant. A daunting hypothesis is, that for PIM, the only needed relation is has Tag expressing that two things are related. This remains to be evaluated.

In general, the approach of the users is to only model when it is necessary and needed later. Participants repeatedly said *I do not want to model the whole world.* Rather, the model is used to explicitly remember important things or facts. As a side-effect this also kept the system usable. A technical limitation of the user interface is that the performance gets worse when many thousand instances are modelled, and the miniquire tree-visualization would then be crowded too much.

From the created classes, we learn that they refine a specific PIMO class and not the generic Thing. Also, the low amount of created classes shows that classes as such are not such an important modeling tool. Removing the ability to create classes at all may remove one burden of the user to decide when to create a class. Also, the selection of classes identified by [12] as useful for PIM were affirmed by the structures created by our participants. Only geographic location was not used as much as expected.

A problem is, that *classes* cannot be annotated like instances can. A class cannot have relations to things, nor can they be annotated in the wiki or be used as Tags for <u>occurrences</u>. So classes are excluded from most editing functionalities. Users wanted to annotate classes to add meta-information about the class. Given the class <u>Requirements Specification</u> (a subclass of Document, instances are concrete requirements specification reports from customers), the user wanted to relate this class with a document that has instructions how to write a <u>Requirements Specification</u>. The same happened with <u>Survey</u> where there are documents about "doing surveys" in parallel to the instances of the class <u>Survey</u> as such.

The contextual inquiry also influenced the participant and the data as such. At the beginning of the interview, one participant noted that *"this should be part of that"* and modified his PIMO. The behaviour of users doing organization work during interviews was also noticed by Barreau and Nardi [1]. Key results of the 2006 case-study were:

– The *drop-box* component increased productivity as it allows to file items faster than without the assistance.
– The possibility to add multiple categories to a document was used, in the mean 2.5 categories were attached to a file, which is significantly more than the single category a hierarchical file system provides.
– The participants agreed that the PIMO reflects their personal mental models.
– The *gnowsis desktop search* was used very frequently and users found unexpected information.

It is interesting to note that the two long-term users who were interviewed in 2008 did not use the desktop search frequently. The key patterns learned from the interview in 2008 are:

– The ontology tree-view presented in the miniquire user interface is the major entry point to the data. This confirms the results of [1] where users also browsed first and only when not finding elements used search.
– The semantic wiki feature is crucial. The text notes were used daily and for various tasks. Users found creative ways to realize task management and note-taking.
– Applications which are not yet integrated are described as a problem, and a hindrance to use the Semantic Desktop. Plugins for all desktop applications are needed to further support users.
– The PIMO structures enable the users to replicate their mental models using associations and multi-criterial classification [4]. The ability to structure information helped users to creatively file their information and remember elements. They used the structures as entry point to their files.

Both *methods* used for evaluation, the long term case study and the contextual inquiry brought insight about how users cope with Semantic Technology. Clearly, more experience is needed to improve the quality of the studies, here a cooperation with usability experts would be fruitful. Besides that, the technical

effort is not to be underestimated. For example, participants had the possibility to integrate their MS-Outlook e-mails, contacts, and appointments with gnowsis, but didn't use this option because installation was too complicated and they feared that bugs could damage their data. Hence, software problems will always influence your evaluation.

5   Future Work

Looking at the *limited* results that were given by the questionnaire and the evaluation logging, and the *rich* information about practical experiences retrieved by video-recording contextual interviews, more evaluations with interviews could be done. Another evaluation with the NEPOMUK prototype is planned for mid-2008. The question is how the Semantic Desktop and PIMO compare to existing file systems and software. With the deployment of KDE 4.0 in July 2008, the Semantic Desktop and the idea of the PIMO will be delivered to more than a million users, which is partly a success of our work and the Semantic Web, but also opens a challenging field for research. For KDE, we reported the issues found in the long-term contextual inquiry to the main developer Sebastian Trüg, who is addressing them there.

Acknowledgements

References

1. Barreau, D., Nardi, B.A.: Finding and reminding: File organization from the desktop (1995)
2. Boardman, R.: Improving Tool Support for Personal InformationManagement. PhD thesis (July 13, 2004)
3. Cheyer, A., Park, J., Giuli, R.: IRIS: Integrate. Relate. Infer. Share. In: Proc. of Semantic Desktop WS at ISWC (2005)
4. Dengel, A.R.: Six Thousand Words about Multi-Perspective Personal Document Management. In: Proc. EDM IEEE Workshop. IEEE, Los Alamitos (2006)
5. Gemmell, J., Bell, G., Lueder, R., Drucker, S., Wong, C.: Mylifebits: Fulfilling the memex vision. In: ACM Multimedia, France (December 2002)

---

[11] `http://nepomuk.semanticdesktop.org/`

6. Greenberg, S., Buxton, B.: Usability evaluation considered harmful (some of the time). In: Proc. CHI 2008, pp. 111–120. ACM, New York (2008)
7. Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., Gudjonsdottir, R.: The NEPOMUK Project — On the way to the Social Semantic Desktop. In: Pellegrini, T., Schaffert, S. (eds.) Proceedings of I-Semantics 2007, JUCS, pp. 201–211 (2007)
8. Heim, D.: Semantic Wikis in knowledge management - Evaluating the Gnowsis approach. Master's thesis, Fachhochschule Kaiserslautern (August 2006)
9. Jones, W., Teevan, J. (eds.): Personal Information Management, October 2007. University of Washington Press (2007)
10. Kelly, D.: Evaluating personal information management behaviors and tools. Communications of the ACM 49(1), 84–86 (2006)
11. Kiesel, M.: Kaukolu: Hub of the semantic corporate intranet. In: Proc. of the Workshop SemWiki 2006 - From Wiki to Semantics at the ESWC Conference (2006)
12. Latif, K., Tjoa, A.M.: Combining context ontology and landmarks for personal information management. In: Proc. ICOCI, Kuala Lumpur, Malaysia (June 2006)
13. Morse, E.L.: Evaluation methodologies for information management systems. D-Lib Magazine 8(9) (September 2002)
14. Nielsen, J.: Why you only need to test five users (March 2000),
http://www.useit.com/alertbox/20000319.html
15. Quan, D.: Designing End User Information Environments Built on Semistructured Data Models. PhD thesis, MIT (2003)
16. Sauermann, L., Bernardi, A., Dengel, A.: Overview and Outlook on the Semantic Desktop. In: Decker, S., Park, J., Quan, D., Sauermann, L. (eds.) Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference (2005)
17. Sauermann, L., Grimnes, G.A., Kiesel, M., Fluit, C., Maus, H., Heim, D., Nadeem, D., Horak, B., Dengel, A.: Semantic Desktop 2.0: The Gnowsis Experience. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 887–900. Springer, Heidelberg (2006)
18. Sauermann, L., van Elst, L., Dengel, A.: PIMO - a Framework for Representing Personal Information Models. In: Proc. I-Semantics 2007. JUCS (2007)
19. Schuhmacher, K., Sintek, M., Sauermann, L.: Combining Metadata and Document Search with Spreading Activation for Semantic Desktop Search. In: Proc. of ESWC (to appear, 2008)