Gunnar Aastrand Grimnes, Benjamin Adrian, Sven Schwarz, Heiko Maus, Kinga Schumacher
und Leo Sauermann

Semantic Desktop for the End-User

Semantic Desktop für Anwender

semantic desktop_usability_recommendations_information integration

Summary. This article describes the Semantic Desktop. We give insights into the core services that aim to improve personal knowledge management on the desktop. We describe these core components of our Semantic Desktop system and give evaluation results. Results of a long-term study reveal effects of using the Semantic Desktop on personal knowledge work.

Zusammenfassung. Der Semantic Desktop ist ein Ansatz um persönliches Wissensmanagement zu unterstützen. In diesem Artikel werden der Semantic Desktop und seine Dienste vorgestellt. Ergebnisse einer Langzeitstudie zeigen, dass einfache semantische Relationen bereits großen Nutzen für das persönliche Wissensmanagement bringen.

1. Introduction

Modern desktop environments offer a range of applications for storing personal information of every conceivable kind. However, the data is kept isolated in single applications, and it becomes the responsibility of the user to be aware of where and how she stored each bit of her information, and to keep the integrated view of the information in memory.

The Semantic Desktop aims to solve this problem of information integration and makes personal information management (PIM) on the desktop more powerful. It uses Semantic Web technologies to interrelate, annotate and categorize all the information found on the typical desktop. The user is free to relate information as she wishes and is given modeling tools for representing her own ideas of the world on the computer. The Semantic Desktop allows the user to free herself from a static folder structure. Instead she can organize her data by any desired dimension, by relating and integrating data items across application boundaries (Sauermann *et al.*, 2009).

We envisage a Semantic Desktop to be a gradual improvement on current desktop technologies, a unifying and integrating layer on top of existing desktop applications. Furthermore, the semantic layer allows for realizing intelligent assistants considering the user's personal view and experience. Thus, for our research the Semantic Desktop is a base technology to support knowledge work and finally, contributes to the organizational knowledge management.

1.1 Benefits for the end-user

The core motivation for the Semantic Desktop initiative is improving PIM on the desktop. In particular our Semantic Desktop solution aims to address a set of prototypical office tasks that are very time intensive or hard to solve on non-semantic desktops i. e.:

Representing Things and Their Connections

Although modern desktops offer applications to work with a wide range of different information resources, from addressbook and calendar entries to songs and images, there is no way to represent or store general concepts. Which application would you use to work with *research ideas, contacts I met at CeBit,* or *Project-proposal brainstorming meetings* and many other abstract or fleeting concepts? The Semantic Desktop lets the end-user model a personal ontology where these types of things and many others can be represented. A user may have classes for people, projects, topics, ideas, etc. These *things* can be related to each other, i. e. *Dirk is also interested in the CID Project,* and are used to annotate and integrate the information from all the other applications. Moreover, the Semantic Desktop environment fosters the emergence of this personal ontology without requiring an intense up-front modeling.

Improved Navigation

By realizing and formally representing implicit connections between bits of information on the desktop, these relations can later be followed and finding related information becomes easier. For example, a meeting concept can be linked directly to: the Outlook calendar entry for the meeting, the people attending, the web-page providing directions for the venue, the booking confirmation for transport and accommodation, the PDF file of the meeting agenda, etc. The additional links of the Semantic Desktop allow the user to explore her data along any dimension she likes. Whereas a folder hierarchy in a file-system is restricted

DOI 10.1524/icom.2009.0033

to a single hierarchy, with the Semantic Desktop she can explore different views as appropriate to the task at hand (e.g., viewing documents by the time they were created, by the persons who worked on them, or by the project they are related to).

It has been observed by (Teevan *et al.,* 2004) that users actually do prefer navigating over search-engines. They show a behavior dubbed *orienteerin'*. Typical steps include: opening and examining an entry point to folder structure (e.g., the *My documents folder*), examining the sub-folders and searching for the right location until the desired item is found. We observed the same behavior with long-term users (Sauermann *et al.,* 2008) that confirm results from end-user studies (Bergmann *et al.,* 2008). They indicate that, even with modern desktop search-engines at hand, users prefer and fall back to navigate in folder structures for re-finding information.

Improved Search

Desktop search-engines like Google Desktop allow fast and efficient keyword searching of the information from all applications. However, such keyword searches break down when the user cannot remember the correct term to use, or the term has multiple meanings. The Semantic Desktop allows much more sophisticated searches, where the relational structure of the data is taken into consideration. This allows the user to search even when she does not remember the exact terms. Semantic Searching also allows searching directly for a fact, not just for a document that may contain the information.

Tagging Documents & Crossing Application-boundaries

On a traditional desktop each application is a fenced off area for information, i.e. although the desktop information space may contain an address-book entry for a person, e-mails from this person, documents where this person is the author, and meetings where this person attended; it is usually not possible to navigate from one to the other. Suites of applications such as Microsoft Office, or the Apple suite of application go some way to solve this, but the solution is restricted to included applications. The Semantic Desktop offers a single integrated information-space where the information from all existing applications can be imported, and the links between them can be established.

Notes on Everything

Only very few applications offer the user the option of taking free-form notes for all information handled, i.e. although it is possible in Outlook to write additional notes about a contact in the address-book, it is not possible in Word or Excel to make additional notes that are not directly a part of the documents being created. By using a personal Wiki and allowing wiki-articles to be associated with every information element on the desktop, the user can write free-form notes about anything (i.e., using the concepts directly in the text). By using a Semantic Wiki, the transition from casual written notes to a formal representation is quick and easy.

Context & Task Management

Many professionals find themselves working in many different roles during the day, e.g., scientists might find themselves spending parts of the day working on industry projects, working on personal research projects, teaching and doing paper reviews. Each of these contexts will have different concepts and documents that are relevant to the task at hand. Manually closing and re-opening the correct applications causes a large overhead when changing contexts often. By explicitly representing work context and the elements of the Semantic Desktop that are connected to each, the user may effortlessly switch between contexts and have the appropriate resources at their disposal. Such explicit context can also be used to tune automatic features to only make suggestions that are currently valid.

1.2 Semantic Desktop Services

Technically the core of Semantic Desktop is the usage of Semantic Web technologies on the desktop, this includes the usage of formal ontologies for describing domains,[1] the Resource Description Framework (RDF) for representing the knowledge,[2] and the query language SPARQL for querying information.[3] In (Sauermann *et al.,* 2006) we presented the basic architecture of a semantic desktop system. The core services are: a service to crawl data on the desktop and convert it to RDF (the Aperture[4] framework), store data in an RDF database, and infer new knowledge from the data. A search service provides full-text and semantic search, a semantic wiki[5] provides means to store text, and an ontology service provides a programmatic API to tag documents or manipulate classes and instances. In the NEPOMUK project we

Figure 1

[1] See http://www.semanticdesktop.org/ontologies/
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/TR/rdf-sparql-query/
[4] http://aperture.sourceforge.net
[5] http://kaukoluwiki.opendfki.de

developed and standardized ontologies for files, e-mails and other document types. Higher level ontologies represent the mental model of the user. The Personal Information Model (PIMO) is a model to represent a single users' concepts, such as projects, tasks, contacts, organizations, allowing files, e-mails, and other resources of interest to the user to be categorized, independent of application and with multiple relations (Dengel, 2006). Based on the core services and the PIMO ontologies, various applications exist to help the user filing and finding information. The core application in the NEPOMUK prototype is the Personal SEmantic Workbench (PSEW), this is the entry point for the user to manage her PIMO ontology, she can navigate and modify the class hierarchy, and annotate and create new relations between things. In Figure 1 we show a typical PSEW session: on the top left is the class navigator; in the middle the thing editor for modifying a single concept together with a Semantic Wiki is shown; at the side the sidebar with the relations to other concepts. In the lower left corner, concept recommendations for the currently selected thing are presented.

In addition to PSEW, the NEPOMUK prototype offers a range of plugins for widely used applications, such as Microsoft Outlook, Mozilla Thunderbird and Firefox, etc. To help the user file new information correctly in her semantic desktop we also offer a DropBox application (see Figure 2). The DropBox helps (semi-) automatically moving and classifying files. It consists of a normal folder called *DropBox* that is observed by the system. When a file is dropped, a window appears showing recommendations for tags to classify the file, and possible folders where to move it. This allows the user to correctly tag documents and move them to an appropriate folder with a single click.

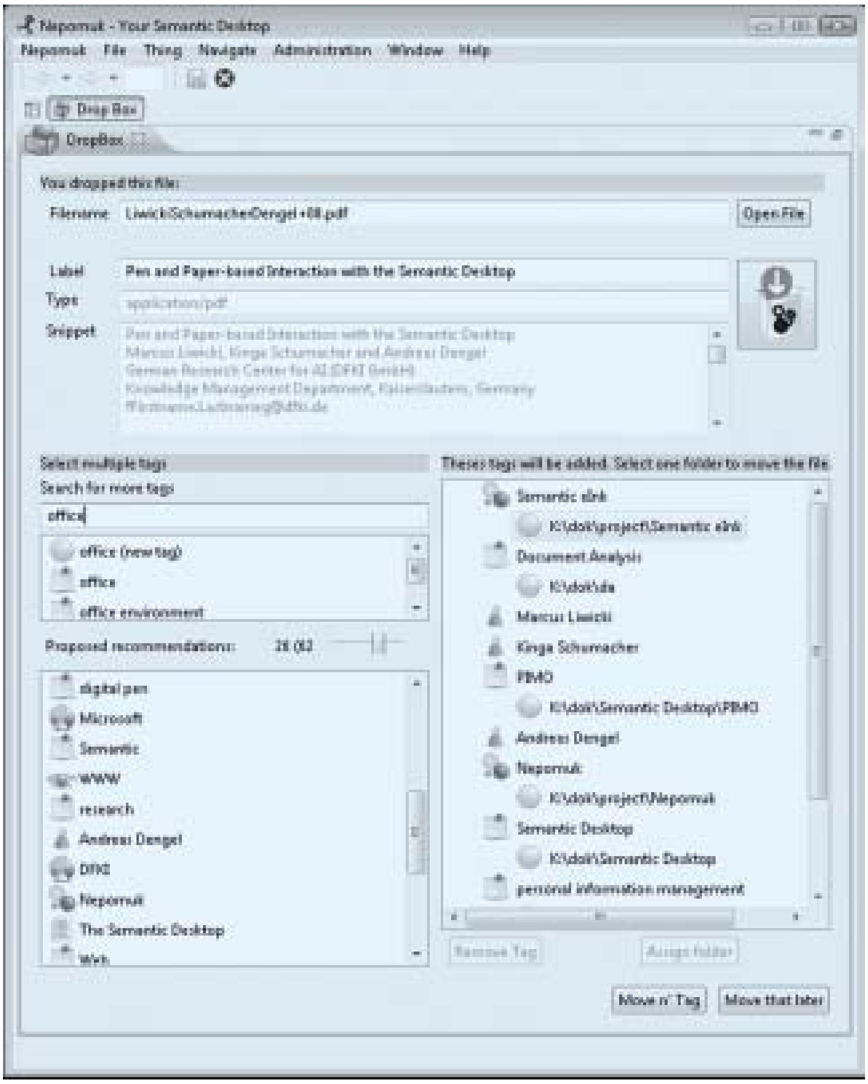

Figure 2

2. Related Work

The Semantic Desktop ideas have attracted a lot of research attention, and many different approaches exist.

The **Haystack** (Quan *et al.,* 2003) project presents a good example for an integrated approach to the SSD field. Inter-application barriers are avoided by simply replacing these applications with Haystack's own word processor, e-mail client, image manipulation, instant messaging, etc. Having all applications built on the same back-end allows end-users to access all possible functions of an object by activating the object's context menu regardless of the currently visible application. In (Quan, 2003, pp.179–183), an end-user study with four participants over four weeks is presented. Haystack is evaluated for its use in PIM, showing that the use of collections is rated positive, support for task-management is rated borderline.

Another similar PIM tool is the **Semex System** (SEMantic EXplorer) (Dong and Halevy, 2005). Semex leverages the PIM environment to support on-the-fly integration of personal and public data. Information sources are related to the personal ontology through a set of mappings. When users are faced with an information integration task, Semex aids them by trying to leverage data collected from previous tasks performed by the user or by others. Hence, the effort expended by one user later benefits others. The algorithm matching multiple entities (in that case, persons) in different data sources is evaluated in a preliminary study, showing that the algorithm and user interface yield a reasonable browsing experience.

A similar idea is exploited by the **IRIS** Semantic Desktop ("Integrate. Relate. Infer. Share") (Cheyer *et al.*, 2005), an application framework that enables users to create a *personal map* across their office-related information objects. Like Haystack, inter-application barriers do not exists, because all applications are made from scratch for IRIS. Each application is implemented as a set of services and data

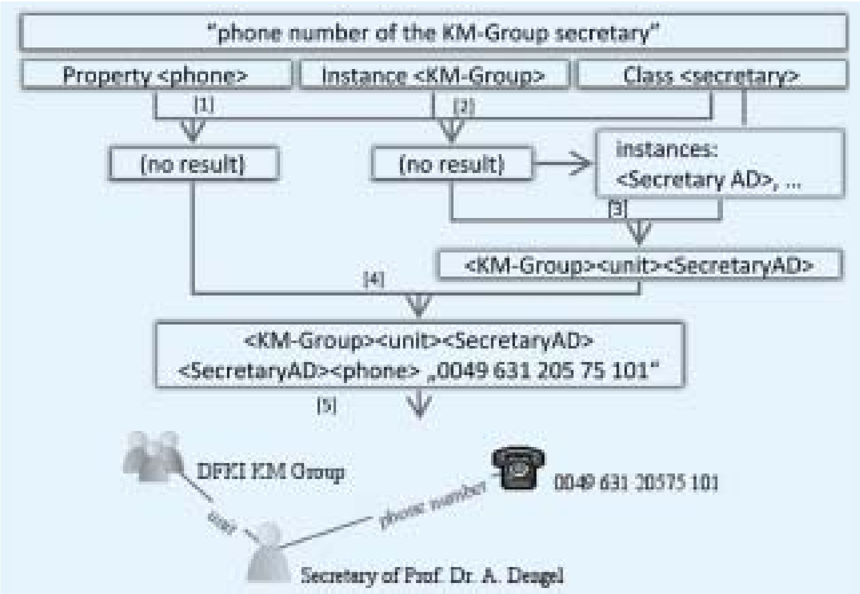

Figure 3

structures, allowing developers to extend the system. A key feature of IRIS omnipresent *recommendations* which suggest annotations for e-mails, appointments, documents, etc. Thus, the user does not have to annotate entities manually. For each element, related elements are found according to text and annotation similarity, allowing users to navigate even without annotating anything. This system was used by 15 users over a few weeks of time, only marginal results are published. One published anecdote is, that the natural language processing of the system reminded a user that there is a *meeting request* in a mail where the user did previously overlook it in the text and forgot to schedule the meeting.

DeepaMehta (Richter *et al.,* 2005) is an open source Semantic Desktop application based on the Topic Maps standard.[6] The DeepaMehta UI, which runs through a Web browser, renders Topic Maps as a graph, similar to concept maps. Information of any kind as well as relations between information items can be displayed and edited in the same space. The user is no longer interacting with files and programs. To our knowledge, no formal evaluation is published.

[6] ISO/I1EC 13250:2003: http://www.y12.doe.gov/sgml/sc34/document/0129.pdf

3. Core components

Our semantic desktop is a framework combining many different technologies to finally create a unified end-user experience. In this section we describe briefly some of the individual parts of our semantic desktop system that implement basic use cases such as search, tag or relate, contextualize, browse and finally take notes.

3.1 Semantic Search

Once all the user's data and documents are collected together in a single knowledge base it becomes possible search the whole information space with a single query, and the high structure of the information allows explicit semantic queries such as: "Find the phone number of the secretary of the Knowledge Management Group". Traditionally, semantic knowledge bases are searched using semantic query languages such as SPARQL, but this requires the user to both know the SPARQL language, the RDF representational language as well as the details of the ontologies used to represent the data. To overcome this problem we devised an interface where natural language queries are transparently and automatically mapped to formal semantic queries. In (Schumacher *et al.,* 2008) we describe a comprehensive semantic search solution which supports semantic teleporting (i.e searching directly for a fact such as a phone number will deliver exactly this fact, not just the document it is contained in), and exploration of the knowledge base to find related documents and things when the user is not exactly sure what is being sought. Our technique uses natural language processing to disambiguate the words used in the query and map these to formal semantic concepts. Spreading activation is used to traverse the semantic knowledge graph and finds the most relevant nodes. In Figure 3, the process is shown. In general, first (1,2) the individual terms of the query are interpreted as subjects, predicates, and objects of RDF statements and corresponding data is retrieved. This first set of results is vague, as individual terms are queried and not the whole phrase matches. In an iterative process, attempts are made to create a chain of connected facts based on the query. In the first iteration (3), an overlap of instances of the class "secretary" and items related to "the KM group" yield the "person who is the secretary of the KM group" as a result binding for those two queries. In the next iteration, a fact is found that this person has a "phone number", so all query terms can be combined into a meaningful chain of facts. This chain is visualized.

We evaluated our approach using details about people attending conference and the papers presented at that conference. We issued a set of typical queries to our own system as well as to Google Desktop Search which had indexed the same data. When querying for simple keyword terms such as "social networks" or simple names "Sintek" the two approaches show similar performance, but our semantic search is superior when doing more complicated queries like "organizers of workshop no. 5" or "DFKI members at the ESWC2007". The evaluation results demonstrate the power of our combined approach; the engine returns immediate precise results to extensive queries by exploiting facts, and also showed that it is appropriate to search for information on the Semantic Desktop in a goal-directed way.

3.2 Rule-based Semantic Search

The search algorithm above does find results in many cases needed for knowledge work. In specific domains, with more knowledge about the end-user and the given tasks, it is possible to improve the search results further by adding domain-specific rules. As starting point we took our semantic desktop search engine which combines search results from the crawled text corpora with the RDF database (e.g., a document found also has associated concepts). It provides a keyword-based search where results are categorized in the main PIMO classes showed on top of the result page and lists further results one the page as usual (see Figure 4). Thus, a user can start a full-text query and access the documents in the result list as she is used to, but has the chance to spot concepts or relations from the PIMO which are missing in classic desktop search engines.

We extended this search with a forward-chaining rule engine and SPARQL queries, and published an evaluation in (Sibert *et al.,* 2006). In this experiment, specific rules are modelled for Siemens Business Services, a consulting branch company. The rules add related documents to search results, in specific cases. For example, when a project is in the search results, the project manager should be also included. In the user interface in Figure 4, a search for “vito” returned both the person “Vito Giannella” and the concept “BRITE”. An analysis of the search results revealed that BRITE actually refers to the project “Business Register Interoperability Throughout Europe”, causing the project to be included. Once the project is included, the project manager “Bertin Klein” is also included. End-user studies showed that these improvements must be explained to the user — in Figure 4, explanations are displayed below the search results, and a “grey” font indicates indirect search results.

Figure 4

3.3 Semantic Tagging

The ontology-based information extraction system iDocument is used to generate recommendations for information objects in the Semantic Desktop and returns a weighted list of document classification candidates. It is currently embedded in two places: First, as a component providing recommendations to the currently active thing editor (see Figure 1, lower left corner) and second within the above mentioned DropBox (see Figure 2, lower left corner).

To see its effectiveness, we present iDocument in (Adrian *et al.,* 2009) where its recommendations were evaluated on real PIMO data. Here, five NEPOMUK users agreed to provide their PIMO model and ten pre-categorized documents. We separated iDocument’s recommendations into four buckets with thresholds in steps of 0.25 from 0.0 to 1.0. Each bucket contained recommended instances if the confidence was greater than the bucket’s threshold. Then we analyzed precision ratios for each bucket. Figure 5 reveals that four users accepted more recommendations in buckets with higher thresholds. This confirmed the quality of iDocument’s result ranking.

3.4 Work Context

Knowledge workers are engaged in multiple tasks. They have to execute these tasks in parallel (Gonzáles and Mark, 2004). Back in 1995 O’Conaill and Frohlich carried out an important and famous user study. Two workers have been shadowed with a video camera for a full working week. The video data revealed that the subjects were being interrupted about four times per hour. The interruptions as well as the recovering cause significant overhead costs. The workers do not only have to remember the former task state. They also need to reconfigure their workspace, allocate relevant resources such as documents, websites, e-mails and contact addresses as well as to launch applications required to resume the task. And, as such, switching to a different task also implies switching to a different set of resources and tools.

The Semantic Desktop provides means to represent and store a user’s *contexts*. Each such context representation references the resources that the user created or consulted in that specific context. The Semantic Desktop keeps track of a user’s current context and aims at reducing the disruptive effect of interruptions by easing transitions from one context to another. To do so, the user’s operations with the computer are observed using an application plugin (see the User Observation Hub project[7]). Interpreting these user actions as a continuous stream of contextual evidences, a *context identification* component analyzes the most recently observed user operations, extracts relevant resources, and searches for the context representation that matches them best. If the *identified* (best matching) context is actually the currently *active* user context, its representation is enriched with the recently touched resources. Otherwise, the system might have detected a context switch and proposes a

[7] http://usercontext.opendfki.de/wiki/UserObservationHub

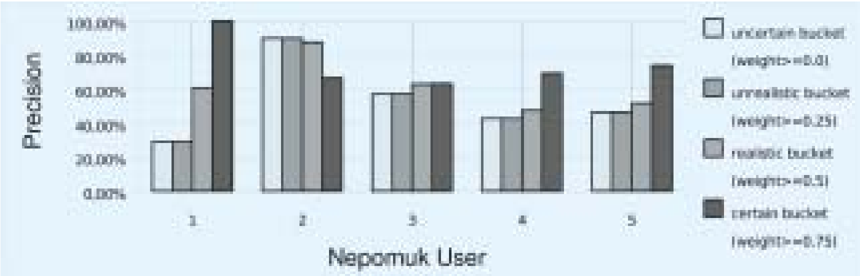

Figure 5

transition to the best matching context to the user. The user is not forced to accept or even react on this proposal, but as long as the system does not get any feedback, it stops to train the context representations. On the other hand, when the user actually wants to switch to a different context, the system assists him with a user interface that looks and feels like an extension of the commonly known multi-desktop paradigm (Schwarz *et al.,* 2008).

Aiming at providing an added value for the user rather than creating even more additional work, it is essential to come up with a robust context identification. The hypothesis is that the context identification estimates the correct context in most of the cases. We evaluated this using a cross validation on a ground truth, manually created by tagging a large set of user operations with one specific "context". A ten-fold cross validation on this ground truth data shows that 78 % of the operations are identified correctly, 9 % of the guesses were incorrect, and 13 % of the cases were not identified at all. Striving for a best-effort strategy, a relatively high number of unidentified cases (13 %) is not considered harmful for the user's actual work. An amount of 9 % incorrect context guesses is not very high, but this is a critical value as false identifications may lead to false context switch proposals and, hence, to disruptions of the user. The reasons for the false identifications are, first, some essential user operations are not being recognized yet by the current user observation software. Additionally, some users specified and separated some contexts that were technically identical (identical set of contextrelevant resources). One way to reduce the latter problem is, again, the development of additional sensors that allow a technical separation of these contexts. In (Dellmuth *et al.,* 2009) first steps towards a more comprehensive coverage of a knowledge worker's environment are taken by extending the user observation to the physical desktop. Here, a digital camera observes adding, removing, and piling paper documents on the desktop. Images taken from the documents are analyzed and fed into the User Observation Hub. Efforts are undertaken to identify if recognized documents are already in the PIMO and introducing them to the PIMO otherwise.

4. Long Term Usability

Although it is useful to evaluate individual components to determine their strengths and weaknesses, the real test of a Semantic Desktop system comes from evaluation of long-term usage of the complete system. (Sauermann and Heim, 2008) describes such a long term study, here several participants used the system for their daily work for a period of two months, two of the users kept using the system and we also describe their experiences after using the system for a period of two years.

For the initial two month evaluation, 8 employees of DFKI volunteered to participate. They were available for the contextual inquiry interviews at the end and for the usability test at the beginning. A team approach, as introduced by (Morse 2002), was used throughout the evaluation. This means, accompanying an evaluation with sessions during which the users exchange their experiences with the software, they explain to each other how to use the software and shared their experiences on how to solve practical problems. 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.* One of the most important questions 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?

4.1 Initial Two-month Evaluation Results

The results were divided into positive and negative feedback. The positive feedback concentrated on the auto-complete functionality of the wiki, and the Semantic Thing Editor. This was followed by the drag-drop functionalities, *starring* things, and the easy installer.

Negative feedback focused on the slow search and the inability to stop a search once started. Users also wanted to filter the ontology tree in the PIMO interface. This was added later and turned out to be a key feature for the two long-term users.

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.

4.2 Contextual Interviews April 2008

Nearly two years after the first evaluation, two users from the original group were still available and kept using the system. They were interviewed in a contextual inquiry in April 2008. Both participants described themselves humorously as *nit-picking information keepers,* one used the German term *Strukturierungszwang*. Both were also involved with the development of the system and should be seen as *"eat your own dogfood"* users, that are biased positively towards the system. As *procedure* we concentrated on a contextual inquiry, as this method brought the most interesting results in the first study. Instead of taking written notes we recorded a video of the interview. We also made a copy of the *Evaluation Logger* logfile, containing all activities the users have been doing over the years.

4.3 Results from the Long-term Evaluation

The two users created only a small number of custom classes and an even smaller amount of custom properties; this 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 sideeffect this also kept the system usable. A technical limitation of the user interface is that the performance gets worse when many thousand instances are modeled, and the ontology tree-visualization would then be too crowded.

5. Future Work

As the Semantic Desktop is the base technology for our research, different threads continue the work on improving the end-user experience of the Semantic Desktop. For instance, we are currently working on integrating an eye-tracker with the Semantic Wiki system, allowing automatic annotations of which parts of a text are read often, and which are merely skimmed. In a collaborative setting this can help other users quickly focus the important parts of large documents. Furthermore, in (Schumacher *et al.* 2009) we present the interplay between paper and the Semantic Desktop by allowing user annotations on paper documents using digital pens and integrating these annotations in the user's PIMO.

Finally, we also aim to improve the quality and usage of automatic recommendations throughout the system, as the overhead of semantic modeling is often prohibitive. With the deployment of KDE 4.0 in July 2008, the Semantic Desktop and the idea of the PIMO was first delivered to more than a million users, and each KDE4 release continue to update and improve the Semantic Desktop features.This is partly a success of our work and the Semantic Web, but also opens a challenging field for research.

Acknowledgements

This work has been supported by the European Union IST fund (Grant FP6-027705, Project NEPOMUK) and is proceeded by the BMBF in Project Perspecting (Grant 01IW08002).

References

Adrian, B.; Klinkigt, M.; Maus, H.; Dengel, A: Using idocument for document categorization in nepomuk social semantic desktop. In: Pellegrini, T.; editor, *I-Semantics: Proceedings of International Conference on Semantic Systems 2009,* JUCS, 2009.

Bergman, O.; Beyth-Marom, R.; Nachmias, R.; Gradovitch, N.; Whittaker, S.: Improved search engines and navigation preference in personal information management. *ACM Trans. Inf. Syst.,* 26 (4): 1–24, September 2008. Article 20.

Cheyer, A.; Park, J.; Giuli, R.: IRIS: Integrate. Relate. Infer. Share. In: Decker, S., Park, J., Qua, D. and Sauermann, L., editors, *Proc. of Semantic Desktop Workshop at the ISWC,* Galway, Ireland, November 6, volume 175, November 2005.

Dellmuth, S.; Maus, H.; Dengel, A.: Supporting knowledge work by observing paper-based activities on the physical desktop. In: *3. Int. Workshop on Camera Based Document Analysis and Recognition (CBDAR 09).* Proceedings, 2009.

Dengel, A. R.: Six Thousand Words about Multi-Perspective Personal Document Management. In: *Proc. EDM IEEE Workshop.* IEEE, Oct 2006.

Dong, X.; Halevy, A. Y.: A platform for personal information management and integration. In: *Proc. of the CIDR Conference,* pages 119–130, 2005.

González, V. M.; Mark, G.: Constant, constant, multi-tasking craziness: managing multiple working spheres. In: *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems,* pages 113–120, New York, NY, USA, 2004. ACM.

Morse, E. L.: Evaluation methodologies for information management systems. *D-Lib Magazine,* 8 (9), September 2002.

O'Conaill, B. and Frohlich, D.: Timespace in the workplace: dealing with interruptions. In: *CHI 95 Conference Companion,* pages 262–263, 1995.

Quan, D.: Designing End User Information Environments Built on Semistructured Data Models. *PhD thesis,* MIT, 2003.

Quan, D.; Huynh, D.; Karger, D. R.. Haystack: A platform for authoring end user semantic web applications. In: *International Semantic Web Conference,* pages 738–753, 2003.

Richter, J.; Vikel, M.; Haller, H.: Deepamehta a semantic desktop. In: Decker, S., Park, J., Quan, D. and Sauermann, L., editors, *Proc. of Semantic Desktop Workshop at the ISWC,* Galway, Ireland, November 6, volume 175, November 2005.

Sauermann, L., Bernardi, A.; Dengel, A.: Overview and Outlook on the Semantic Desktop. In: Decker, S.; Park, J.; Quan, D.; Sauermann, L., editors: *Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference,* volume 175 of CEUR Workshop Proceedings, pages pp. 1–19. CEUR-WS, November 2005. http:// CEUR-WS.org/Vol-175/.

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: *Proc. of the ISWC Conference,* volume 4273/2006 of Lecture Notes in Computer Science, pages 887–900. Springer Berlin / Heidelberg, Nov 2006. ISSN 0302-9743 (Print) 1611-3349 (Online).

Sauermann, L.; Heim, D.: Evaluating long-term use of the gnowsis semantic desktop for pim. In: *Proc. ISWC Conference,* volume 5318 of LNCS, pages pp 467–482, 2008.

Schuhmacher, K.; Sintek, M.; Sauermann, L.: Combining Fact and Document Retrieval with Spreading Activation for Semantic Desktop Search. In: *Proc. of 5th European Semantic Web Conference ESWC, LNCS,* pages 569–583. Springer, 2008.

Schumacher, K.; Liwicki, M.; Dengel, A.: A paper-based technology for personal knowledge management. In: *Proceedings of the 5th Conference on Professional Knowledge Management,* volume P-145 of LNI, pages 289–298, 2009.

Schwarz, S.; Kiesel, M.; van Elst, L.: Adapting the multi-desktop paradigm towards a multi-context interface. In: Kofod-Petersen, A.; Cassens, J.; Leake, D.; Zacarias, M., editors: *HCP 2008 Proceedings, Part II, MRC 2008 – Fifth International Workshop on Modelling and Reasoning in Context,* pages 63–74. TELECOM Bretagne, June 2008.

Siebert, M.; Smits, P.; Sauermann, L.; Dengel, A.: Increasing search quality with the semantic desktop in proposal development. In: *Proceedings of the Practical Aspects of Knowledge Management PAKM conference,* volume 4333/2006 of Lecture Notes in Computer Science, pages 279–290. Springer Berlin / Heidelberg, 2006.

Teevan, J.; Alvarado, C.; Ackerman, M. S.; Karger, D. R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems,* pages 415–422, New York, NY, USA, 2004. ACM.


1


2


3


4


5


6

1 Dr Gunnar Aastrand Grimnes has a Bachelor Degree from the University in Aberdeen, UK, where he also completed his PhD titled "A goal-directed learning agent for the Semantic Web". His research interests include the Semantic Web, Semantic Desktop, Machine Learning and making intelligent software-agents to take over the world.

E-Mail: grimnes@dfki.uni-kl.de

2 Benjamin Adrian studied Applied Computer Science at the University of Kaiserslautern from 2001 to 2006. He graduated 2006 with his Diploma Thesis about a Semantic Tag Recommendation System for the Semantic Desktop. Since 2006 he has been working as PHD student at DFKI, Knowledge Management Department in Kaiserslautern. His research is about semi-supervised learning methods for Ontology-based Information Extraction.

E-Mail: Benjamin.Adrian@dfki.de

3 Sven Schwarz works in the Knowledge Management Department of DFKI GmbH and is responsible for R&D on technically feasible automatisms for analyzing metadata and user interaction streams to infer higher-level conceptualization of the user's resources and contexts. He recently submitted his PhD thesis on "Context-Awareness and Context-Sensitive Interfaces for Knowledge Work Support", which concentrates on the application and utilization of context-awareness to realize unobtrusive, context-sensitive assistance services for knowledge workers.

E-Mail: schwarz@dfki.uni-kl.de

4 Dr Heiko Maus has a Diploma in Computer Science and Mathematics from Saarland University, and a PhD from Kaiserslautern University with the topic "Using workflow context for the realization of process-oriented assistance in organisational memories".

His research combines the fields of knowledge management, organisational memories and ontologies, context research, CSCW, workflow and task management as well as personal information management to realize intelligent assistant systems.

E-Mail: Heiko.Maus@dfki.de

5 Kinga Schumacher completed her degree in computer sciences at the University of Mannheim in February 2006. In her diploma thesis she developed four methods for supervised word sense disambiguation based on statistical analysis and machine learning algorithms. Since 2006, she works as a researcher at DFKI, Knowledge Management Group with a focus on Semantic Search.

E-Mail: kinga.schumacher@dfki.de

6 Leo Sauermann created in 2003 the award-winning diploma thesis "gnowsis", merging Personal Information Management with Semantic Web technologies. He continued work on the Semantic Desktop in the European NEPOMUK project at DFKI. In 2009, he founded the company www.gnowsis.com to provide a commercial product. Sauermann was member of the W3C SWEO Group, and is often invited to speak about Personal Information Management and the Semantic Web.

Privately, the believing Christian enjoys netculture.

E-Mail: leo.sauermann@dfki.de