The Semantic Desktop at Work: Interlinking Notes

Laura Drăgan
Digital Enterprise Research Institute, National University of Ireland, Galway

Siegfried Handschuh
Digital Enterprise Research Institute, National University of Ireland, Galway

Stefan Decker
Digital Enterprise Research Institute, National University of Ireland, Galway

firstname.lastname@deri.org

ABSTRACT
The Semantic Desktop has been proposed as a solution to the ever growing problem of information overload on our computers. It provides the foundations necessary to integrate and manage personal information. However, the challenge of designing and realizing semantic applications that use this infrastructure still remains. In this paper, we present SemNotes, a semantic note-taking tool for the Nepomuk-KDE Semantic Desktop. It provides a real-world, functional use case for fully exploiting the capabilities of the Semantic Desktop: interlinking, organization and management of personal information, improved search and browsing. Furthermore, it represents our solution to a set of identified generic challenges for applications on the Semantic Desktop. We describe a task-based user evaluation comparing SemNotes with a conventional note-taking tool. The results show that complex searches on interlinked information created with SemNotes are significantly faster, with little or no extra effort required from the users.

Categories and Subject Descriptors
I.2.4 [**Knowledge Representation Formalisms and Methods**]: Semantic networks; H.3.4 [**Systems and Software**]: Information networks

1. INTRODUCTION
We accumulate much information on the desktop, so much that it becomes daunting and sometimes useless because of the impossibility to remember it all: people, emails, appointments, tasks, documents, conversations. This information is created and stored in various applications, each with its own data storage, and each with its own format for the data. But in reality this information is not separate, everything is interconnected by implicit or explicit connections. When we use information we do not use just one detached piece of information, but we use data from multiple sources. For example, during a telephone conversation with a project partner, we will need documents about the project, meeting minutes, names of other people involved. Information is inherently interlinked and used in an integrated fashion.

The Semantic Desktop emerged as a solution to the information overload problem on the desktop. It provides a foundation, a framework based on Semantic Web standards, for interlinking information on the desktop. Interlinked information is easier to manage and organize, and because it mimics the mental models of the users, it is easier to find when required. The Semantic Desktop overcomes limitations of the conventional desktops, where information is stuck behind different formats and locked in application repositories. It realizes this by using common vocabularies to describe the data, and a desktop-central place to store it. Thus, the Semantic Desktop creates a pool of linked desktop data, in a standardised format, accessible to all applications.

Indeed, the Semantic Desktop makes the information load manageable. However, new challenges emerge: one such new issue is how to design and realize semantic applications that use the infrastructure provided by the Semantic Desktop. We address this problem by dividing it into smaller, simpler challenges and providing solutions for each of them. To illustrate the solutions, we describe the design of a semantic note-taking application for the Nepomuk-KDE Semantic Desktop, called SemNotes[1]. We use note-taking as an example because it is a desktop activity that is not limited to a specific domain, as the notes can widely vary in topic. It is also a common activity that plays an important role in personal information management (PIM) and that we believe would benefit from the use of semantic technologies.

1.1 Challenges
We present here the challenges we identified as part of the bigger question of how to design new semantic applications for the Semantic Desktop.

- How to use *heterogeneous* data in an uniform way? We refer here to the existing desktop data, which although represented in a standardized way, is not always consistent, up-to-date or even correct. Furthermore, the data is also heterogeneous in terms of form: there are new types of information available to be integrated (i.e. tags, relations).
- How to *interlink* information items? This challenge is in fact the most important one for adhering to the

[1] `http://smile.deri.ie/projects/semn`

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

Semantic Desktop requirements of creating and maintaining a network of linked desktop data. It refers firstly to the new information coming into the desktop through the application, and how to integrate it with the linked information network that exists on the desktop. But it also covers the situation when the only new information created is in fact a new link between existing entities.

- How to design *interfaces* to support the existing workflow of the user while integrating the additional information? This is a question of usability of the application. A balance must be found between too much information, so that it interferes with the user doing his tasks, and too little, or not enough to make a difference.

- How to correctly *evaluate* the application? The scarcity or obscurity of semantic applications for the Semantic Desktop, means that the best candidates for a comparative evaluation are conventional (i.e. not semantic) applications with similar functionality. For task based evaluations, it is difficult to find a common set of tasks so that both applications can perform. A solution is to choose general, high level tasks, but that influences the granularity of the results. Another challenge related to evaluation is the lack of a standardized dataset to use, due to the highly personal data required.

2. THE SEMANTIC DESKTOP

We begin with a few notes on the Semantic Desktop, as it provides the background of our work.

There are several Semantic Desktops described in the literature, different in realization, but all with the same goals: enabling the users to perform their (knowledge work) tasks more efficiently, by making "infoglut" manageable. Envisioned by Vannevar Bush, the Memex [3] has influenced most of the Semantic Desktops, past and current. The most famous is Haystack [10]—a versatile system that served as a semantic browser, desktop and publishing platform—emphasizing the importance of linking information and providing a highly flexible user interface and data model. Gnowsis [14] is one of the first implementations of a Semantic Desktop. It proposed an architecture based on a local Web server as a desktop service, and a generic personal information model for the data. IRIS [4] is a semantic interface to the DARPA funded Artificial Intelligence engine CALO, providing a plugin-based framework for personal information integration. NEPOMUK [1] defines a reference architecture based on Semantic Web standards adapted to the desktop environment, and a set of *desktop ontologies*. The list of Semantic Desktops continues with systems including SEMEX [5], X-COSIM [7], Personal Radar—the predecessor of Twine.com, and The Chandler Project[2].

They all propose extending the conventional desktop with Semantic Web technologies that allow interlinking data that otherwise would be locked in application specific formats and repositories. A common architecture is based on a central repository for the semantic data, which is described using

[2]http://osafoundation.org

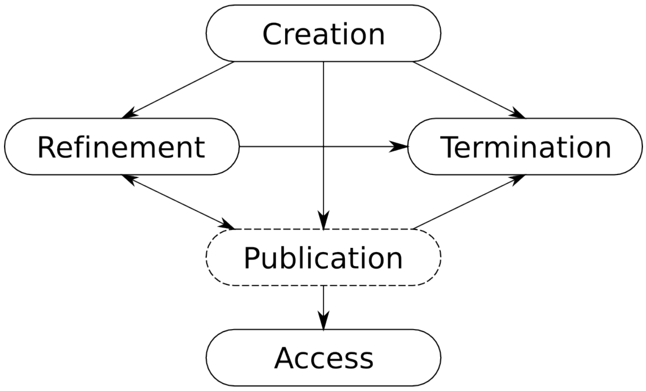

Figure 1: Semantic Desktop Data life-cycle.

one or more predefined ontologies, and which is extracted from desktop information sources like emails or the file system. Most of the systems offer services over the semantic data, like inference, annotation and query. One of the goals of the NEPOMUK project was to create a blueprint for a generic Semantic Desktop architecture. The result is a layered architecture, based on a set of core services, the central one being the desktop storage service. On top of the services lay the end-user applications.

For SemNotes we chose Nepomuk-KDE[3], a branch of the NEPOMUK project, because of our familiarity with the system, but mostly because it is open-source and it has an active community of users and developers.

2.1 Semantic Data

The data on a Semantic Desktop can be of two types:

Vocabulary data. This type of data is rarely handled by the users, and is modified even more rarely, as shown by the long term experiment by Sauermann [15]. A set of predefined vocabularies are loaded by the system, and other vocabularies can be loaded manually by the users, if needed.

Instance data. This type of data is the focus of the paper. Instance data is the RDF version of the users' desktop data. It can be produced and modified: i) *automatically* by crawlers that parse and transform desktop data from standard file formats to RDF; ii) *semi-automatically* through end-user applications; and iii) *manually*, by directly handling triples in the repository—a use case for power users.

Both types of data are stored in the central repository, and are accessible to all applications via the storage service. The distinction between them is made using named graphs of different types—*Ontology*, *InstanceBase* or *KnowledgeBase*.

2.2 Semantic Desktop Data Life-Cycle

We adapted the Abstract Data Life-Cycle Model [13] to illustrate a comprehensive workflow for the semantic desktop data, from its creation to its termination. Figure 1 shows the phases and transitions between them, focusing on the possible actions that the user can execute, and hence, that

[3]http://nepomuk.kde.org

a semantic application could support. Application design should take into account that users are not (and should not have to be) familiar with the inner workings of semantics, therefore all details about ontologies should be hidden from view, although available to power users. The users should be able maintain full control over their data—having full access to change and delete it at any time.

Creation. Most often creation implies creating a new resource, of a given type. However, the import of existing data from other applications or formats (e.g. crawlers—as discussed above) is also included here.

Refinement. The activities specific to this phase include any that make changes to existing data. This phase also contains the creation and deletion of links between existing resources, although alternatively they can be included in the Creation or Termination phases, as links are data too.

Access. This phase is represented by accessing the data through either query or browsing. Because we are discussing interlinked data, accessing a piece of semantic data implies recursively accessing the sub-graph of resources semantically related to it. How much of the sub-graph is traversed can vary, and further traversal by the user should be supported and encouraged.

Termination. It represents the phase when the data is deleted from the system. As with the access phase, rules must be predefined to determine how much of the dataset a deletion will affect—e.g. it might make sense to delete all the subtasks of a task when the parent is deleted, but it does not to also delete the documents related to the task.

Publication. This phase represents making the data accessible to users from outside the system. Also included here is exporting the data to other formats and applications. When handling semantic personal data, applications should ensure that sensitive data is well protected against unauthorized or accidental publication.

3. TACKLING THE CHALLENGES: THE DESIGN OF SEMNOTES

We use a specific application, SemNotes, to describe the general principles we used for the design of an application for the Semantic Desktop. Note-taking is a good use case for the Semantic Desktop because the information contained in notes is not restricted to a specific domain. Personal notes are naturally connected to the user context, and can thus be meaningfully interconnected with the existing network of personal data on the desktop.

We divided the design into specialized modules. Each module handles a set of related tasks. We describe here the modules as they would be for a general application, followed by a more specific description about SemNotes in the next section. The modules are:

Data representation. This module handles the vocabulary of the application: the *data types* around which the application is centred, and the kind of *metadata* that is needed for them. To enable the rest of the Semantic Desktop applications to understand the data produced, as well as our application to understand data from the underlying Semantic Desktop, it is best practice to reuse the existing desktop ontologies where possible. The basic data type handled by SemNotes is the note, represented by a short snippet of text containing personal information.

Data management. This module manages the life-cycle of the semantic data that the application handles, by enabling the transition between the phases described in Section 2.2. Furthermore, this module handles where and how the data is *stored*. The Semantic Desktop provides the framework for storing semantic data, therefore it is best that the central desktop repository be used, when practical. This enables easier interlinking with the rest of the data. In the case of SemNotes, the notes and all the metadata about them are indeed stored in the desktop repository.

Interlinking. This module is logically a sub-module of the previous one, as it specifically manages a part of the *Refinement* phase of the data life-cycle. However, because it tackles the important second challenge listed in Section 1.1, we describe it as a standalone module. The interlinking module effectively realizes the goal of integrating the new semantic data into the pool of existing linked desktop data. The functionalities offered can vary from simple automatic linking of new resources to a specified context or to their author, to complex extraction and inference of new relations and resources. This module provides the feature that sets SemNotes apart from other note-taking tools, the interlinking of the notes with the desktop resources mentioned in them. In our application, there are two sub-modules, that handle (i) entity recognition, and (ii) information extraction, suggesting possible relations to be created by the user.

Visualization. The visualization module presents the data to the user in a simple, yet useful and versatile way. It addresses the third challenge listed above: designing the interface. Depending on the application, the visualization can include *aggregated* views on the data, and *filters*. Faceted search [19] has proven useful for semantic data, and it can be used to present in a meaningful way the interlinking information to the user. In SemNotes, the data that needs to be visualized is basically an enhanced version of a list of notes, with sorting and filtering. The module also manages how the recommendations for interlinking are presented, and the note editor.

We describe how we tackled the last challenge listed—the evaluation of the application—in Section 5.

4. IMPLEMENTATION OF SEMNOTES

In the development of SemNotes, we tried to reuse as much as possible of the features provided by the host Semantic Desktop, Nepomuk-KDE. Using the existing functionality enabled better integration with the rest of the Semantic Desktop, as well as reduced the effort required for the implementation. Nepomuk-KDE provides out of the box central RDF storage for the desktop, and efficient means to access and query the data. It also provides a set of vocabularies collectively called *the Nepomuk desktop ontologies*[4]. The most relevant is the Personal Information Model (PIMO)

[4] `http://www.semanticdesktop.org/ontologies`

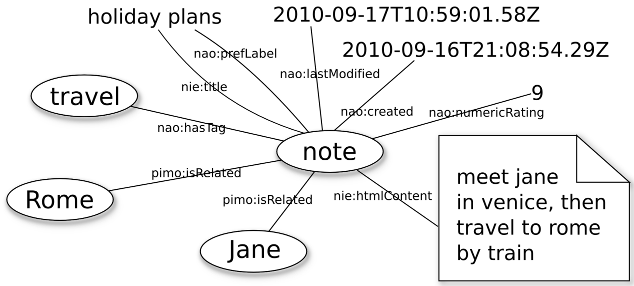

Figure 2: Metadata about a note.

which describes the entities and relations that are "within the attention of the user and needed for knowledge work or private use" [16].

4.1 Data Representation

We describe the data using a subset of the desktop ontologies—PIMO, Nepomuk Annotation Ontology (NAO), and Nepomuk Information Element (NIE). The note is the central unit of information handled by SemNotes—it is represented as an instance of `pimo:Note`. The metadata stored for each note consists of: a title (`nie:title`), creation (`nao:created`) and last modification (`nao:lastModified`) dates and tags (`nao:hasTag`). Figure 2 shows a basic example.

In this first implementation we use tags as only categorisation mechanism, preferring simplicity over the more accurate mix of categories, topics and tags. Because notes are generally short [2] we decided to store the content in the RDF repository, as a property of the note (`nie:htmlContent`). The value is the HTML string representing the content of the note, including formatting. This decision enables us to use the indexing and full text search feature provided by Nepomuk-KDE.

As we discussed, the most important feature of SemNotes is the interlinking of notes with relevant resources from the desktop. The relations are stored using `pimo:isRelated`. In the current revision of SemNotes, this is the only type of relation. This decision was based on results of the long term study of Gnowsis usage [15] which suggested that for PIM tasks it is enough to express that two things are related, and that the simpler properties are preferred by users over the more specific ones, regardless of the possible loss of meaning. However, we consider extending the range of possible relations in future versions. Having the information about the resources that are linked, specifically about the type of the resources, we can infer the possible relations to suggest, based also on knowledge from the desktop ontologies.

Restricting the types of possible relations also keeps the interface simple, which was one of the design goals, and one of the bigger challenges we encountered, as we show in more detail in 4.4. We further explain the extraction and creation of relations in 4.3.

4.2 Data Management

SemNotes supports all the phases of the note life-cycle, as described in Section 2.1. When a new note is created, a new URI is generated for it, and the creation time is set. The rest of the properties are not set at creation time. As note data is added (i.e. the refinement phase), the metadata about the note and the content is updated. At each new update or annotation of the note, the last modification date is also updated. The notes can be found (i.e. the access phase) through full-text search, filtering by tags, related resources, and by creation date. Once the required note is found, it can be viewed in the editor. When a note is deleted, all metadata and relations about it are deleted as well, however, none of the tags and related resources are removed. The publication phase is also supported by SemNotes, as notes can be exported as HTML or text files, or even published online as blog posts [6].

As mentioned above, SemNotes uses the RDF repository provided by Nepomuk-KDE for all data storage.

4.3 Interlinking

The interlinking of notes with related resources is the key feature of SemNotes. It is what realizes the actual integration of the new information with the existing network of linked desktop data. Annotating the notes with related information captures the context of the note. Context is important for personal information management because it enables reminding and better (more precise, faster) search. Links between resources also support *wayfinding* [9] and encourage exploratory browsing and serendipity.

The module uses entity recognition and string matching algorithms to detect and suggest possible related resources, but no link is created until the user selects the correct one. This mixed-initiative approach is a compromise between the precision of the links created and the amount of interference with the user's workflow.

The current implementation suggests annotations based on the knowledge about existing desktop resources, using entity matching techniques to identify the likely candidates. Certain types of resources are more likely to be related to notes (e.g. people, organizations, projects, events, tasks, other notes, locations). By default, SemNotes restricts the search for suggested resources to these types, but the user can easily modify the list. We do not include every resource on the desktop because of the large number of files that are indexed by the Semantic Desktop, which would clutter the suggestion list. For the resources of the given types, all textual properties are compared against the note text. This way, resources that do not explicitly have the search term in their label will show up in the suggestion list. An example is shown in Figure 4: for the word "John" are suggested other notes that mention him, even though the name is not present in the label.

SemNotes does not currently offer suggestions based on online resources[5], unless there was a desktop resource previously created for the relevant Web data (i.e. a bookmarked Web page becomes a desktop resource).
We are working on an information extractor module, which identifies new information in the content of the notes. It will suggest the creation of *new* desktop resources from the text, like events, tasks or contacts, that will then be linked to the

[5]We envision functionality similar to `www.zemanta.com`.

note.

The annotation suggestions are computed while the user types the note, therefore efficient processing is required. The process of finding possible matches follows:

1. Scan the text and identify possible entities represented by a single word or a sequence of words.
2. For each possible entity found in the text find a list of existing desktop resources that match it. We use string matching to compute a score for each resource found. The score takes into account the length of the matched string, and if the resource has been linked to the note before.
3. Sort the matches by score and present them to the user in a non intrusive way (see next section).
4. If the user chooses any of the presented suggestions
    - create the link between the piece of text identified as an entity and the actual resource it represents.
    - use the selected suggestion in the recalculation of the scores for the entities found for the rest of the note. Once a note is linked to a resource, that resource is more likely to appear again, and therefore it will be ranked higher.
5. If the user ignores the presented suggestions, no links are created, but the possible matches are saved for later use.

For the purpose of establishing context, and organization of notes, it is sufficient to create a single link between a note and a resource it is related to, regardless of how many times that resource is mentioned in the content of the note. Therefore the relation between the note and the desktop resource is created only once in the repository. However, if the note is viewed in SemNotes, all the links that the user created to the related resource are displayed.

The interlinking module also manages removal of links between notes and desktop resources. Because the suggestion of related resources is based on the content of the notes, once the last textual link to a related resource is removed, so is the relation between the note and the respective resource.

SemNotes also supports the manual creation of links between the note and the desktop resources that are relevant but have no explicit mention in the text.

4.4 Visualization

SemNotes displays the notes as a list that can be sorted by title, creation or last modification dates or rating. Each note can be opened in-list for quick access, or maximized over the entire window, for viewing and editing. In the in-list mode, several notes can be open and edited at once.

Adjacent to the list of notes, aggregated views can be displayed or hidden. An aggregated view on the list of notes is based on a restricted set of properties that the notes have in common. Depending on the set of properties used (one or more), the most suitable visual representation of the aggregated view can vary. Currently SemNotes offers a tag cloud, a timeline and a related-resource view, each aggregating information about the notes based on a single property (i.e. the tags associated, the creation date and the related resources, respectively).

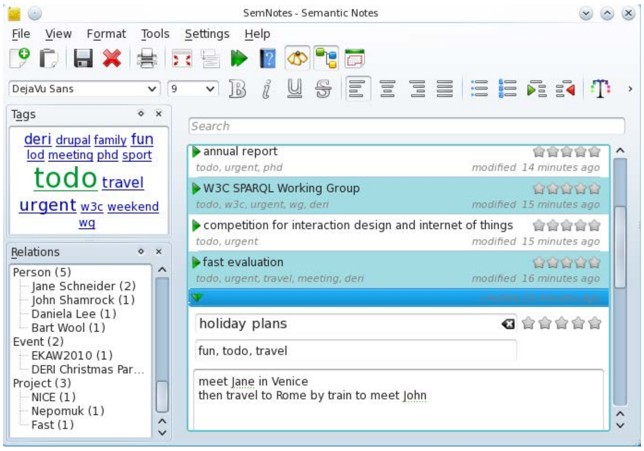

Figure 3: SemNotes GUI.

These views also act as a custom faceted browser, as they provide filters on the list of notes. Filtering the list is as easy as clicking on a tag, time interval or related resource. Multiple filters can be set at once, of mixed types. Figure 3 shows SemNotes with the tag cloud and related-resources views visible, and a tag filter set. A note is open in-list for editing. For more screenshots see `http://smile.deri.ie/projects/semn`.

The editor component provides rich text editing of the note content, and easy editing of the note metadata. Tagging provides auto-completion based on all the tags on the Semantic Desktop, and creating a new tag is transparent to the user. If the user does not set a title, SemNotes automatically sets it to the first line of the note.

The suggestions for annotations are presented in a simple non-intrusive way, in the "spell-checker" style (i.e. the words for which suggestions were found are underlined with a green dotted line), and are available as context menu items, by right-clicking. Figure 4 depicts how annotation suggestions are presented, and how a linked resource is displayed in the note. To remove an annotation is just as easy as creating it—through a context menu item.

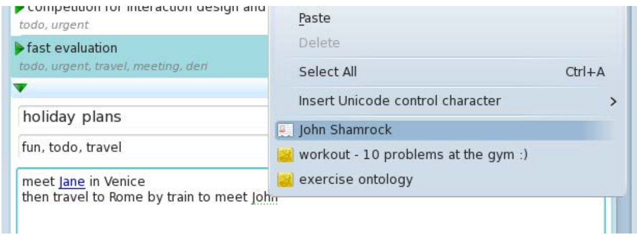

Figure 4: SemNotes annotation suggestion and link.

5. EVALUATION

We conducted a task-based user evaluation, comparing SemNotes to the popular note-taking application Evernote. The goal of the experiment was to determine if the effort spent for the creation of links between notes and resources is repaid by easier search. Towards this goal, we measured the time spent executing the same set of tasks with both tools, and compared the results. After the experiment, we asked the participants about their experience with SemNotes in a questionnaire.

The two applications we compared have the same main functionality, note-taking. We chose Evernote as baseline, as it is a very popular note-taking tool that is freely available. Its set of functionalities are richer than those of SemNotes, but the basic features are present and similar in both applications. The feature that distinguishes SemNotes is the same that we want to measure—the creation of links between the notes and desktop resources, based on suggestions given by the application. SemNotes runs on KDE on Linux, and uses the framework provided by Nepomuk-KDE Semantic Desktop. Evernote runs on several operating systems for desktop and mobiles, but Linux is not one of them, therefore we used its Windows version.

Participants. Twenty participants took part in the evaluation, all members of our research institute. They are students and researchers in the field of Semantic Web, thus possibly favouring the semantic application. This bias (if it exists) would however influence only their responses in the questionnaire and not the times measured. Fourteen of them regularly use note-taking and five use Evernote as their preferred note-taking tool. None of the participants used Linux as their operating system of choice. Their familiarity with one environment and one application could have influenced the measured results, in favour of Evernote.

Some additional demographic details of the participants: gender distribution was even (ten men and ten women); most of them are in the age groups between 25 and 29 (eleven, equivalent to 55%), and between 30 and 34 (five, or 25%). There are seven Master students, seven PhD students, five senior researchers and one intern.

Data. We used two virtual machines for the experiment, preloaded with identical data. We chose data familiar to the participants, which they use in their everyday work. The dataset contained contact information for 130 members of our institute; 655 recent emails from our mailing lists; 20 scientific papers authored by our colleagues; and photos from institute events. The note data was also identical: 50 notes on a variety of topics, personal or work-related, tagged with 23 tags. In SemNotes, we also provided links between the notes and the resources mentioned in them: people, projects, events or other notes.

Tasks. We prepared a set of eight tasks. Each participant was requested to run all the tasks in each of the two environments. To prevent order effects from influencing the results, half of the participants started with SemNotes and the other half with Evernote. The first two tasks were intended to help the participants get accustomed with the data available and the environment; we did not include the time spent on these tasks in the results.

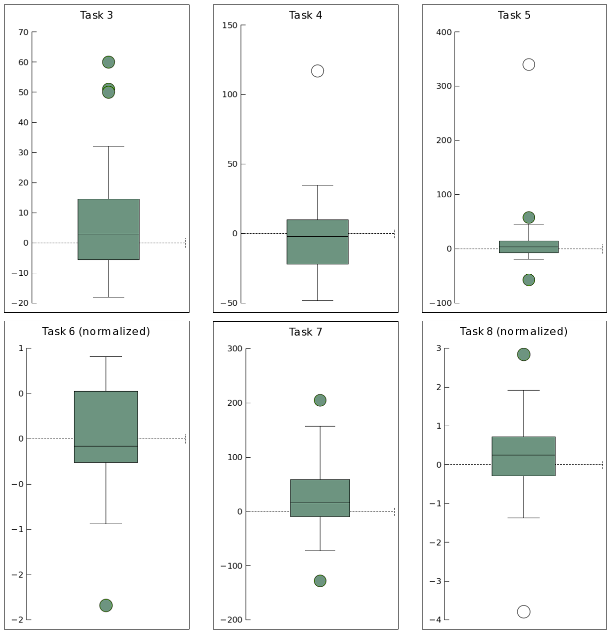

Figure 5: Inter-quartile ranges for the time diff.

The other six tasks were focused on note-taking, and their complexity increased gradually.
T3. Find notes tagged with “todo”.
T4. Find to-dos that are related to our institute.
T5. Find a to-do related to a presentation by a colleague John.
T6. Take a note about planning a social event for your research group. Write the names of two people that have already confirmed. Annotate the note as you see fit.
T7. Find a note containing the minutes from the last meeting about a given project. Change the date of the next meeting planned.
T8. Take a new note for the action item assigned to you at the last meeting of the project. The action item is in the meeting minutes previously edited, and it requires drafting a document using a paper authored by a colleague. Annotate the note.

T3, T4, and T5 are search and filter tasks, from the very simple to the complex. T6 and T8 are editing tasks, including any annotations that the participants made on the notes. T7 is both search and edit task. It prepares the ground for the most complex editing task, T8, which required interaction with the rest of the system (i.e. finding the required paper).

Measurements. For each task we measured the effort in seconds spent, and number of mouse clicks. We also counted the number of key strokes for the tasks involving creation of notes, and we used this measure to normalize the values for time. Although we had two separate measures for each task, we were interested in the difference between these values, computed according to:

$$value_{difference} = value_{SemNotes} - value_{Evernote}$$

Thus, a positive value means that the value recorded for SemNotes is bigger than the corresponding value for Evernote, and a negative value means that the value for SemNotes is smaller.

For the time measurements, the time difference represents the extra *effort* required for annotating the notes with links - in the editing tasks. For the search tasks the difference shows which of the applications enables the users to find the notes easier, thus the *benefit*. We must specify here that actual searches done by the software were considered instantaneous, as the datasets are small and we did not intend to measure the performance of the search algorithms used.

For the number of mouse clicks, the difference represents the extra *effort* required, for both types of tasks.

After a first analysis of the results, we noticed that some values for time were unrealistically high or low. They coincided with the measurements when the participants stopped to ask a question or comment on a feature. We decided to eliminate these outlier values by using only the measurements that fall in the inter-quartile range, for all tasks—see figure 5.

5.1 Quantitative Results

We tested if the effort measured in time and mouse clicks is different when using SemNotes than when using Evernote. Table 1 shows the results, with the statistically significant values in bold ($p < 0.05$).

T	Time			Clicks		
	Avg	Med	*t*	Avg	Med	*t*
T3	0.5	0	0.152	0.167	0	0.692
T4	-8	-8	**-2.94**	-0.333	-1	-0.48
T5	-0.125	1	-0.046	0.857	1	1.426
T6	0.063	0.016	0.486	6.067	8	2.026
T7	14.357	13	1.713	4.812	2	1.527
T8	0.249	0.243	1.004	20.8	12	**3.08**

Table 1: Statistics for time and click differences.

Results show that for the search tasks T3 and T7 the differences are positive, which suggests that it takes longer to finish the tasks with SemNotes. However, they are not significant. For the complex search tasks T4 and T5 the difference has negative values, showing that the users spent less time on complex searches when using SemNotes, thus supporting the claim that the use of interlinked data makes notes easier to find. Only the results for T4 are statistically significant.

For the tasks that required creation of new notes, T6 and T8, the values are positive, thus it took longer to finish the tasks with SemNotes. This was expected, as in SemNotes there is the additional step of annotating the notes with links to desktop resources. However, the differences are small and not significant.

Regarding the number of clicks, only task T4 yielded a negative difference, thus reinforcing the result above. For all the other tasks the values were positive. Only task T8, that required creating and annotating a complex note based on information from other sources, had statistical significant difference in number of clicks, in favour of Evernote (more clicks needed in SemNotes). This is not surprising, as the participants recognized the value of creating links and proceeded to link the new note to the relevant resources.

In summary, the results show that there are significant improvements (for one of two tasks) in the time spent on complex searches, when the data is interlinked, at no significant extra cost for the creation of the links. Linking does however significantly increase the effort measured by number of mouse clicks (for one of two tasks).

5.2 Questionnaire

After the evaluation we asked the participants to fill in a questionnaire related to the experiment. According to the answers, on a scale from 1 to 5, the tasks were simple (mean 2.25) and similar to the ones in their daily work (mean 3.4), and the data provided was familiar (mean 3.2).

The answers also show that 60% of the participants (12) felt that SemNotes helped them finish the tasks *faster*, while only 20% (4) considered Evernote, and 20% did not feel any difference between the tools. When asked which of the two applications helped them perform the tasks *better*, 80% of the participants chose SemNotes, while the rest of 20% did not feel that there was any difference.

The participants had a good overall impression of SemNotes (with a mean of 4.15 on a scale from 1 to 5).

6. RELATED WORK

Semantic note-taking means enhancing the note-taking process using Semantic Web technologies. It can refer to the techniques and methods used in the implementation, like ontologies and RDF, but most importantly it is about creating a semantic network around the notes and the information contained in them. There are several applications that enable more or less semantic note-taking, some are browser based (online or offline), while others are standalone desktop applications as is SemNotes.

List.it browser-based note-taking tool [12] and its predecessor Jourknow [11] save context alongside the information scraps, to improve re-finding and reminding. It also features information extraction from the unstructured text of the notes, recognizing entities and relations between them, with the *pidgin* language processor. SnapShoot [8] is another browser based note-taking tool that explores new visualization techniques to improve reading of the documents produced. It features categorization, and limited interlinking with documents within the system.

A distinct category of semantic note-taking applications are personal semantic wikis, like Kaukolu [18], IkeWiki [17] and GDKTiddlyWiki. Each wiki page represents a resource and its semantic relations to other resources are encoded within the page, using an extension to the wiki syntax. Only predefined relations and types are available, and the wikis offer limited access to other desktop information sources.

OneNote from Microsoft's office suite provides quasi-semantic functionality by interlinking the notes with address book information, calendar, and tasks. It does not use any semantic technologies though, and the data is locked in by proprietary formats and storage.

Other note-taking applications that provide semantic features include: Jenga Note[6]—allows associating a note with a concept, Catch Notes[7], and SpringPad[8].

Zemanta[9] is a blog assistant that suggests possible enhancements to blog posts, like linking external content and images. Unlike SemNotes, which uses the local repository to search for matches, Zemanta looks on the Web. It does not assign any semantics to the links.

7. CONCLUSION

In this paper, we presented a solution to the challenge of designing applications for the Semantic Desktop. We described one such application, SemNotes, a semantic note-taking tool for the Nepomuk-KDE Semantic Desktop. It provides a real-world, functional use case for fully exploiting the capabilities of the Semantic Desktop: interlinking, organization and management of personal information, efficient search and browsing. It supports the entire life-cycle of the semantic data represented by notes, with emphasis on the creation of links between the new data and the existing network of linked information on the desktop.

We describe also the task-based user evaluation comparing SemNotes to the popular note-taking application Evernote. The results show that no significant extra effort (measured in time spent) is required for the annotation of notes with links to related resources, but that the benefit (measured in time saved) is significant for one of two complex search tasks.

8. ACKNOWLEDGMENTS

The work presented in this paper was supported by the Líon-2 project funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380.

9. REFERENCES

[1] A. Bernardi, S. Decker, L. van Elst, G. Grimnes, T. Groza, S. H. M. Jazayeri, C. Mesnage, K. Moeller, G. Reif, and M. Sintek. *The Social Semantic Desktop: A New Paradigm Towards Deploying the Semantic Web on the Desktop*. IGI Global, October 2008.

[2] M. Bernstein, M. V. Kleek, D. R. Karger, and m. c. schraefel. Information scraps: How and why information eludes our personal information management tools. *ACM Transactions on Information Systems*, 26(4), 2008.

[3] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[4] A. Cheyer, J. Park, and R. Giuli. Iris: Integrate. relate. infer. share. In *Proceedings of Semantic Desktop Workshop at ISWC 2005*.

[5] X. Dong and A. Y. Halevy. A platform for personal information management and integration. In *Proceedings of CIDR 2005*.

[6] L. Dragan, A. Passant, T. Groza, and S. Handschuh. Linking personal semantic notes. In *Proceedings of KIELD 2010*.

[7] T. Franz, S. Staab, and R. Arndt. The x-cosim integration framework for a seamless semantic desktop. In *Proceedings of K-CAP 2007*.

[8] S. Iga and M. Shinnishi. Snapshoot: integrating semantic analysis and visualization techniques for web-based note taking system. In *Proceedings of APVIS 2006*.

[9] W. Jones. *Keeping Found Things Found: The Study and Practice of Personal Information Management*. Academic Press, December 2007.

[10] D. R. Karger, K. Bakshi, D. Huynh, D. Quan, and V. Sinha. Haystack: A customizable general-purpose information management tool for end users of semistructured data. In *Proceedings of CIDR 2005*.

[11] M. V. Kleek, M. S. Bernstein, D. R. Karger, and m. c. schraefel. Gui — phooey!: the case for text input. In *Proceedings of UIST 2007*.

[12] M. V. Kleek, M. S. Bernstein, K. Panovich, G. G. Vargas, D. R. Karger, and M. M. C. Schraefel. Note to self: examining personal information keeping in a lightweight note-taking tool. In *Proceedings of CHI 2009*.

[13] K. Möller. *Lifecycle Support for Data on the Semantic Web*. PhD thesis, National University of Ireland, Galway, 2009.

[14] L. Sauermann. *The Gnowsis Semantic Desktop approach to Personal Information Management*. PhD thesis, Fachbereich Informatik der Universität Kaiserslautern, 2009.

[15] L. Sauermann and D. Heim. Evaluating long-term use of the gnowsis semantic desktop for pim. In *Proceedings of ISWC 2008*.

[16] L. Sauermann, L. Van Elst, and K. Moeller. Personal information model (pimo) specification v.1.1, 2009.

[17] S. Schaffert, R. Westenthaler, and A. Gruber. Ikewiki: A user-friendly semantic wiki. In *Proceedings of ESWC 2006*.

[18] L. van Elst, M. Kiesel, S. Schwarz, G. Buscher, A. Lauer, and A. Dengel. Contextualized knowledge acquisition in a personal semantic wiki. In *Proceedings of EKAW 2008*.

[19] K.-P. Yee, K. Swearingen, K. Li, and M. A. Hearst. Faceted metadata for image search and browsing. In *Proceedings of CHI 2003*.

[6] `http://www.jenganote.org`
[7] `http://www.catch.com`
[8] `http://www.springpadit.com`
[9] `http://www.zemanta.com`