UNIVERSITY OF GOTHENBURG

Composing an Evaluation Framework for Personal Information Management Tools

KLAS ESPENKRONA
MATTIAS SVENSSON

Bachelor Thesis

Report No. 2009:043
ISSN: 1651-4769

University of Gothenburg
Department of Applied Information Technology
Gothenburg, Sweden, May 2009

Composing an Evaluation Framework for Personal Information Management Tools

Klas Espenkrona
espenkro@ituniv.se

Mattias Svensson
svemat@ituniv.se

ABSTRACT

Many of the employees of today's corporations are knowledge workers, people who create, read, analyze and act upon information. Information that is stored in different places, in different file formats and accessed by different software. PIM (Personal Information Management) literature expresses a lot of challenges for our tools to overcome. We have compiled these into an evaluation framework that could be used to evaluate future PIM solutions. We have also created a software prototype with the purpose of addressing the challenges in the evaluation framework, which we conduct a theoretical evaluation of.

1. INTRODUCTION

Many of the employees of today's corporations are knowledge workers, people who create, read, analyze and act upon information. Digital information is most often located either on the local computer, the company's intranet or the Internet. The main purpose behind Personal Information Management (PIM) is the re-access of already found information that can be situated in any of the three information contexts, which the knowledge workers are forced to shift between many times per day [1]. Each context comes with a set of different tools to access the information, such as email clients and file browsers, which provide their own methods for finding and organizing information. Jadaan and Stenmark suggest that a program that allows for seamless movement across the information contexts would improve PIM in general [1]. A solution would be to build an all new kind of software, uniting all three information contexts. The drawback of this would be software development costs and that a company would have to abandon much of their already working software. We theorize that an interface covering the information contexts would suffice. A thorough review of literature revealed several challenges associated with PIM. This information was spread over several articles. To be able to pin point the challenges to address with a new piece of software some kind of criteria matrix would have to be compiled. In this thesis we aim to create a software prototype that addresses as many of the challenges found in the literature as possible. A theoretical evaluation will then be conducted to how well the prototype handles the problems. The objective of this thesis is thus twofold:

To identify and compile the challenges expressed in PIM literature and to devise a prototype interface that addresses these challenges.

In the next chapter of this thesis the research method used is presented, followed by a chapter containing a literature review of previous research in the field. The next chapter presents the evaluation framework which is a set of criteria by which the solution will be evaluated. In the next chapter the functionality of the prototype is described and ventilated. This is followed by a theoretical evaluation of the prototype using the framework, and then the problems and challenges are discussed along with future research. The final chapter will contain the conclusion and summary of the thesis.

2. RESEARCH METHOD

Since one of the questions we needed to answer was what the actual challenges with PIM are, this research was performed using an exploratory approach. A prototype was then developed and evaluated theoretically against the challenges found.

2.1. Literature review and framework

A thorough literature review was made first of all to obtain necessary knowledge in the area, but also to summarize the problems and challenges that exist in the PIM area today.

We used Google Scholar[1] as our main source of articles about PIM. We decided to use the ten most cited articles on the keywords "personal information management", and all articles since January 2008 on the keywords "'personal information management' refind" as a starting point. That way we got both the most influential and the most recent findings on the subject. After reading the articles we could remove a few that were not so relevant to what we were doing, i.e. not discussing challenges of PIM.

When reading the articles we took notes on the different problems and challenges presented within, even though they might not have been the focus of the article. Some challenges were mentioned in clear text while other were hidden between the lines. The challenges were noted in a spreadsheet and iteratively reorganized into emerging categories. Similar to what grounded theorists call saturation, we continued this process until the categories appeared to be stable and no new challenges appeared.

The list of challenges was then transformed into a framework that could be used to evaluate PIM tools. The framework presents these challenges and only that. Solutions are proposed later on in the prototype section. We hope that the framework could be of some use to software designers so that thoughts might be raised about design decisions and maybe result in new and intuitive solutions to information refinding and organization.

2.2. Prototype and evaluation

The framework is in itself a good contribution to the area of PIM, especially for designers and developers of PIM tools. In addition we also developed a user interface prototype to try to provide solutions to some of the problems. This prototype was then theoretically evaluated against the framework we designed.

The prototype was built using an already existing product called Newton, owned and developed by GlobeAccess[2], as a base. We were meant to develop the prototype by programming it ourselves, but the company decided that they wanted to do it. Our work instead consisted of coming up with features that we felt Newton were lacking, and create concept images and descriptions of these. This limited our chances to actually try out some of our ideas and especially to tweak them towards perfection. The ideas for the prototype were developed together with the framework and worked as a method to enhance the framework. Of course it was also the other way around; the framework (previous research) helped us find out what features Newton was lacking.

Conducting only a theoretical evaluation is obviously a somewhat limited approach to testing software features, since solving a problem in theory does not mean anything has been solved in practice. Testing this in a live environment is however beyond the scope of this thesis. The process of evaluation was somewhat arbitrary. First we would suggest one or more features based on the framework for Newton to GlobeAccess. After some time we would receive the latest version of Newton with the new features in it. Once these were received we would test the new feature(s) in the most common scenarios and then discuss the outcome and conclude on whether the problem was properly addressed. Because we

[1] http://scholar.google.com

[2] http://www.globeaccess.com

lacked access to the source code, tweaking and trying different approaches to a problem was impossible and time did not allow us to resubmit feature suggestions to GlobeAccess as we wanted to cover as many features as possible.

From now on we mean the original application, with its original functionality, when writing "Newton", and Newton with our ideas incorporated when writing "the prototype".

3. LITERATURE REVIEW

In this section previous research on PIM will be reviewed. It has been divided into four subsections, one for each area where PIM is utilized; email, file systems and files, web and special PIM software. The purpose is to introduce how the users use the available software, what problems they have, and what features that might be missing. As we mentioned in the research method section, some of the articles that we read were not fully relevant. These are not presented here. A list of these articles can be found in Appendix A instead.

3.1. Email

Email has been around since the early eighties and is still one of the most popular means of communication over the internet. Research shows that email users tend to use email for more things than just communication such as archiving and task management. Whittaker and Sidner call this "email overload" [3:276] and argue that this creates problems in personal information management as an inbox might contain hundreds of messages with conversations, documents and task reminders in no specific order [3]. Ducheneaut and Belotti state that this mess is created because knowledge workers have a tendency to "embed personal information management directly into their favorite workspaces" [4:30]. In the follow up to Whittaker and Sidner's article Email overload Fisher et al. establish that the most notable change in our email use over the ten years past is that email archives have grown tenfold [5].

Ducheneaut and Belotti also found that when filing email the hierarchies used were often shallow. The reason for this was that the user wanted to be able to quickly access items. Some nesting of folders was merely a reaction to limited screen height since users want certain items visible while they inspect others [4].

A common use of email is for filing information. Jones et al. [9] state that a common way for users to keep track of useful information is to send themselves emails with relevant information along with a comment describing the information, as the name of the file or address was often not enough for the user to determine if the information was useful or not.

3.2. File systems and files

Information stored on the local computer or on file servers is often arranged in files and folders with describing names. Jones et al. state that a folder structure is used not just to store files but could also represent the users emerging understanding of a project and its sub-components, thus the folder becomes information in its own right [2]. Files and folders can, like emails, become "overloaded" for example when the users have to make folder names start with "aa" so that the file browser will show them at the top of the hierarchy. Another problem with a strict hierarchy is that a content item (document, image, etc.) can go in only one place.

Jadaan and Stenmark claim that a large part of the users would rather search manually i.e. use the file browser and navigate to the correct file, than make use of local search tools such as the windows search service [1].

As noted by Boardman and Sasse, users tend to retrieve their email by sorting on metadata, such as sender and date received [13]. Another type of metadata, "tagging", is found useful by Dumais et al. [7]. Both of these also state that folders and file hierarchies might not be necessary because of that.

Jadaan and Stenmark state that some users do not save documents on their local computer that are accessible on the intranet. This is because the information on the intranet might update while the information on the local computer becomes obsolete [1].

3.3. Web

Most web browsers today has the possibility to store URLs for the user, yet this feature is seldom used as the information provided by the browser about the URL often is not enough remind the user what the URL contained. This is stated by both Jones et al. [8] and Jadaan and Stenmark [1].

Jones et al. [9] observed that the same problem that occurs with email and file hierarchies over time, occurs with web bookmarks too. If not maintained, the bookmark folder gets "overloaded".

3.4. Special PIM software

There has been previous research on prototyping of new, special, PIM interfaces. As Boardman and Sasse mention, there have been two types of approaches when trying to improve integration between tools. The first approach is embedding support in an existing tool, i.e. in an email client. The second is a new, unifying, interface for multiple types of information [12].

WebView is a graphical software designed to help the user find their way back to web pages that they have visited. The software is an add-on to the web browser and creates a map, with names and thumbnail pictures, of the pages visited by the user. According to the author's preliminary evaluation the software gave indications of improved efficiency in some navigational tasks [6].

Another piece of software that is developed by PIM authors is "Stuff I've seen" [7] a.k.a. SIS. SIS is a tool that registers all the websites, files, etc. that the user accesses. The user can later use free text to search amongst the already accessed material. This software covers all the information contexts but offers no kind of overview of the information accessed, but simply a hit list similar to that of today's search engines.

Perhaps the most common type of PIM software is the desktop search, which is included in most modern operating systems (such as Windows Search[3]) but also available as stand alone software (for example Google Desktop[4]). The functionality of these is often very similar to that of SIS but instead of only being able to search on items already accessed by the user they index all available information thus making it searchable without prior access.

3.5. Summary

As seen in this chapter a lot of prior research has been done in the field of PIM, most of which focuses on a single problem or challenge. We have summarized these from the different areas in PIM and compiled them into a framework. This framework could be an aid when conducting research or designing new software.

4. EVALUATION FRAMEWORK

In this section of the thesis the evaluation framework that can be used to evaluate PIM tools is presented. The framework is a compilation of the problems and challenges described in the literature review and they are described in a more concrete way here. At the end of the chapter an even more concrete summary of the framework is presented.

The overview, fragmentation, organization, and searching sections were chosen as they match the different types of challenges we found during the literature review. Some overlapping occurs between the different sections though.

4.1. Overview

Dong and Halevy [10] writes that according to Vannevar Bush [13] the human mind does not think by the way of directory hierarchies, but

[3] http://support.microsoft.com/kb/940157/en-us
[4] http://desktop.google.com

rather by following associations between related objects. Today there are not very many possibilities to easily create links between objects and there are very few alternatives for browsing information by relation.

A problem with folder hierarchies is that they can obscure as well as organize according to Jones et al. Information filed away is “out of sight, out of mind and easily forgotten” [2:1505]. As folder hierarchies hide files they reduce the overview the user gets and forces the user to browse to the bottom of the tree to see what the lowest level of folders contain. As folder hierarchies are one of the most common ways to keep information in order many different applications use them, thus forcing the user to switch between them. This is a cause for fragmentation. As users wanted to have fast access to their files folder hierarchies in email inboxes were often shallow, in general no more than two folders deep. Folder hierarchies also offer a somewhat logical view as the information items contained within does not have to be in a certain format to get grouped with related items. For a folder view to be a complete logical view it would have to be able to create links and relations between items and for items to appear in multiple places without being copies.

When information is filed it often lacks context which makes it harder for the user to re-find it with just a glance. File names and URLs are often not descriptive enough to tell the user exactly what they contain, and certainly not for what purpose they were saved. According to Jones et al. users send themselves emails containing URLs along with descriptions of the content of the URL and in what context it is supposed to be used [9].

4.2. Fragmentation

Information fragmentation occurs when related information is spread over different formats, handled by different applications and situated in different contexts (such as the local file system or the email inbox). To find related information the users must launch many applications and perform repetitive searches. Fragmented information also leads to inconsistencies when the users update the information in one place but fails to update it elsewhere [11]. Related information could also be spread over the three information contexts, the local, the intranet and the web. This forces the user switch between these contexts in order to access information and could also increase the inconsistency of the information.

4.3. Organization

According to Ducheneaut & Belotti [4] as well as Jones et al. [2] users can not always sort information items in a preferred way. Users work around this by overloading folder names by giving them names starting with “AA” so that they are placed at the top when sorting by name. When ordering information, e.g. in the inbox, it is often sorted by metadata, such as when it was received, who sent it etc. [14]. As of today there are no standard for tagging and adding metadata to information items independently of its format, thus each application and file format has incorporate this by themselves. More research is needed on this Jones et al. argues [11].

Most application lack support for task management, the exception being some email clients with calendars incorporated. Users work around this by creating folders or files with names that remind them of the task and place it where they see it, such as the desktop. Email is also used frequently for reminders as users address mails to themselves according to Jones et al. [ref]. When users try to incorporate information, about for example due dates, into formats that does not support it, the information item can become “overloaded” [3]. Overloading can cause distraction and obstruct the finding of relevant information due to large amounts of overhead information.

4.4. Searching

Dumais et al. states that between 50 and 80 percent of websites accessed were re-visits of previously accessed pages [7]. Information items marked with tags and metadata increase the list of keywords applying to that object, thus could make it easier to find and re-find relevant information. Searching by concept could also become possible with extensive tagging and metadata marking. Searching for i.e. “environmental disasters” will only result in information that contain these exact words but not relevant information not containing these words. Synonyms and homonyms are also connected to keyword-based searching.

Result lists today tend to be very similar, using the internet search engine hit-list style, used by, amongst others Google. Research has shown that grouping can be very useful when presenting search results [16].

Most users strive to achieve their goals with a minimum amount of cognitive load (the load on short-term memory during thinking and reasoning) and therefore prefer recognition tasks to recall tasks. This means that rather than use a keyword search, users would navigate to information even though it takes longer time. Because it has be done before it is easier for the user to recall when clicking links than figuring out what keyword to use and browse through the results.

4.5. Summary

In this section we will try to summarize the framework in a more concrete way, so that software designers easily can use it to evaluate their own PIM tool. The points that are described below are all the features that we have found through reviewing other literature in the area and that we believe should exist in a PIM tool today. See Table 1 for an example on how we evaluated the prototype we developed. In Table 1 we have also summarized the points below and categorized them in the same way as this chapter was divided into sections.

Relations/Logical view: The human mind wants to browse information by relations rather than folder hierarchies. Many computer users build their hierarchies using relations but the problem with strict hierarchies are that some files need to be copied to multiple places to be allowed to relate to more than one item.

Folder hierarchies: As file hierarchies grow, it gets harder to get an overview of all files on your system. There should be no need to browse back and forth between folder hierarchy levels to find the folder or file you are looking for.

Context: Sometimes users save files for later use. Often they are having trouble remembering which file was which. The user should not have to open the file to find out what it contains or in what context it belongs.

Fragmentation: Information exist in many different sources. In a complete PIM tool, information should be fetchable from many different sources, e.g. from email, the web and the local file system; so that the user does not have to switch between applications to find an item that they forgot where they stored. This does not mean that the PIM tool should be able to open and edit all these different files, only to let the user refind them.

Sorting and filtering: Users often sort and filter data based on metadata such as creation date etc. They should not have to rename folders or files to sort them the way they want to.

Metadata and tagging: Unfortunately there are no standards for defining metadata for a file today. There are for some types of files, e.g. music files, but they also differ between different formats. We believe that the PIM tool should take care of this in some way, but we agree that more research is needed on this [11].

Task management: Calendars, reminders and "to do" lists all come in many different shapes. We suggest that this should be built in the PIM tool and preferably allowed to be connected to other items, for example files and emails connected to a specific deadline.

Searching: A PIM tool of today cannot become popular without search functionality.

The tool should not restrict the user to only searching though, it should still offer browsing through relation hierarchies. Common search engine problems include searching for synonyms/homonyms and how to present the results. This thesis does not talk a lot about these problems but we do want to mention them as we believe that they should be considered when designing search functionality for a PIM tool. We have divided the problems into three categories, presented below.

Keywords: Synonyms/homonyms to keywords should also present the correct results. All metadata/tags should be searchable.

Result list: Does the user get a good overview of all hits? Or how do we know which hits to prioritize?

Cognitive load: Most users does not want to think more than necessary. Can the search functionality help the user by, for instance, suggesting additional keywords?

5. THE PROTOTYPE

As mentioned in the research method section, one part of our research was to develop a prototype in which we would try to incorporate solutions to the problems and challenges we found while doing the literature review. We used an application called Newton as a base for our prototype.

In its original version, Newton only worked with local content. We wanted to see whether Newton could be extended to cover all three information environments - the local, the organizational, and the global.

We thus added support for function calls to web based resources such as search engines, including desktop, intranet and web search. By defining a node as containing the results from a Google Desktop search, a user may collect all local emails matching a project name or a team member's name in a node. A node can also hold a URL to an intranet or a web site, to a shared file server inside the organization or to an RSS feed. This way, the prototype can provide users with the ability to span the local, the organizational, and the global information context via a single interface.

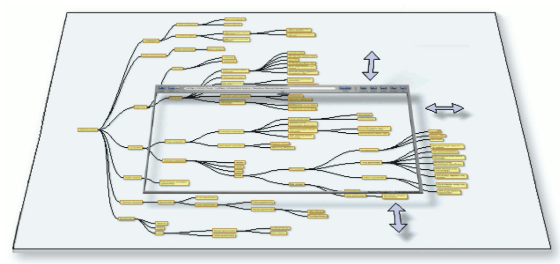


Figure 1. The content and its organization can be overviewed by zooming in and out using the mouse wheel or clicking the map. Scrolling is accomplished by clicking and holding the mouse key while dragging the window over the tree map.

5.1. The Content Map

Newton offers a graphical map-based interface to data, which resembles that of a mind map. The interface consists of two parts. The main part is the Content Map, which is used for navigation and overview, displaying how information objects are logically organized and related. The Content Map consists of nodes that are connected in a tree structure similar to that of traditional folders in a folder hierarchy. A node is like a virtual folder, and can contain files, such as Word documents, Excel files, or images. New nodes can be created from scratch and manually populated with content from different sources. Nodes can also be populated by importing an existing folder structure from the user's local hard drive. A node can also be empty and used only to represent the aggregated content from its sub-nodes.

As in a traditional file system, there can be unlimited levels of sub-nodes. In Newton, however, the nodes can be created, moved, deleted, sorted and re-arranged in any way the user wants without altering the physical location of the objects. The new structure, which is purely logical, is stored in a database. The tree structured map is zoomable and scrollable both vertically and horizontally, letting the user either get a good overview of the whole tree or focus in on a particular node (Fig. 1).

In the prototype we added the ability to create a post it-like node which you can place wherever you want on the Content Map. This could be used as a reminder for a task that you have to perform in, for instance, a project. After implementing that we came up with the idea of creating these reminders as content items instead, to allow the user to create a calendar-looking interface, but that is not implemented yet.

5.2. The Content List

The second part of the graphical user interface contains a detailed list of all the individual items contained in a node. We call it the “Content List”. The Content List is only shown when a node has been selected. However, since nodes often have sub-nodes, the Content List not only shows the items of the selected node but also the aggregated list of all the information objects from the sub-nodes, sorted per sub-node (Fig. 2). The sub-node labels can be toggled on or off using the View button on the top menu. With this option turned off, all objects are displayed in a single list.

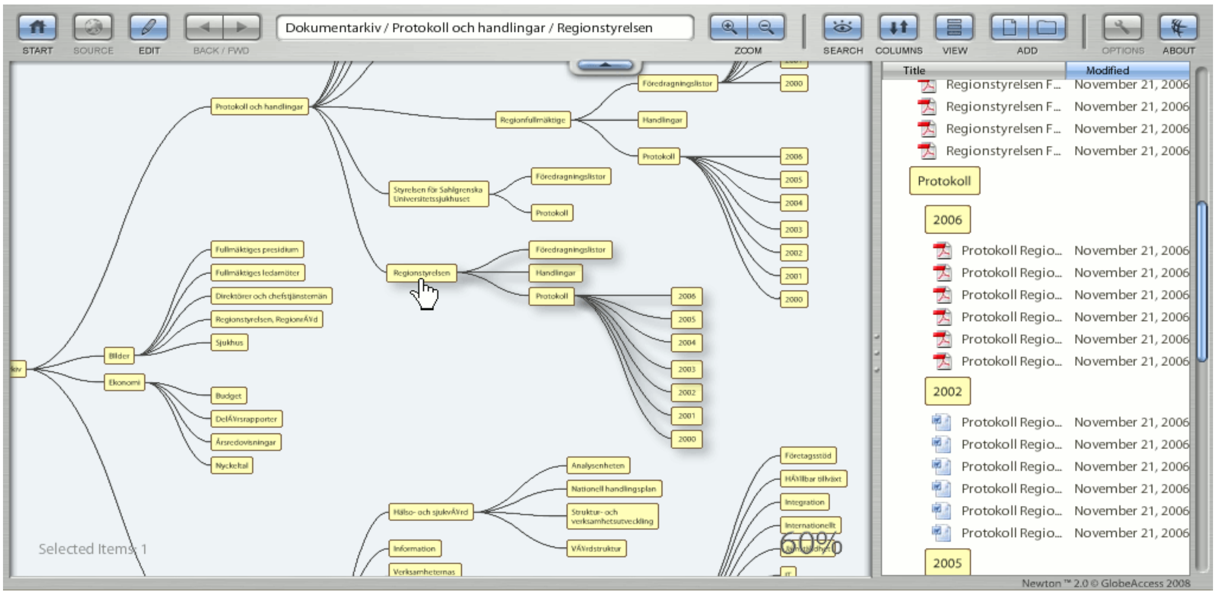
In Newton, the items in the Content List were always sorted on their name. In the prototype we added the possibility to sort on other metadata, such as item type and item modification date, which was the only supported metadata at the time.

The only other metadata we added support for was comments, and we felt that was enough as it allows users to tag its content in any way they want. Each content item can have a comment connected to it to help the user remember what specific files and web links actually are, without having to open them. The comments pop up automatically as the user moves the mouse over an item. Automatic metadata extraction, e.g. the sender of an email, is outside the scope of this thesis.

All types of files can be added to a node, and the same file can (logically) be added to several different nodes. The user can find all instances of a particular file by selecting it in the Content List and looking at the Content Map for the attention icon (Fig. 3). In the prototype this method can also be used to find objects based on item type or modification date.

The titles of the nodes and the content items can be searched from within Newton. The results are presented by displaying in which nodes there are hits, similar to how multiple instances of a selected content item is shown in figure 3. This is an area in which Newton have

Figure 2. Once a node is selected by a mouse click, the Content List is opened at the right hand side, showing the names and content of all sub-nodes.



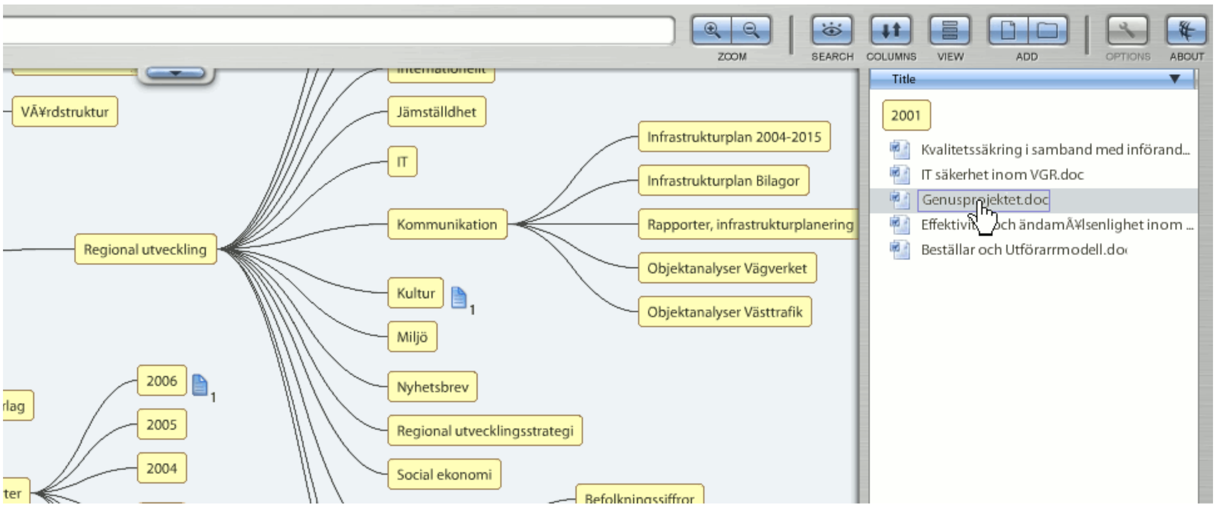


Figure 3. Clicking on an object in the Content List, the blue attention icon shows where in the tree the selected document is located. If there are multiple instances, as in this case, all occurrences are indicated.

to develop to become a truly great and usable PIM tool, perhaps by indexing everything that is added and allow the user to search within this index.

6. THEORETICAL EVALUATION

In this section a theoretical evaluation on the prototype will be presented. Please note that the prototype is not a finished application and the solutions to the problems and challenges identified by the framework are suggestions to solutions. These suggested solutions were created by us to the best of our knowledge.

We are evaluating both Newton and the prototype. As the prototype is merely an extension of Newton all the features in Newton are still intact in the prototype. Thus, when stating that a solution applies to Newton it also applies to the prototype, but not the other way around. The prototype has features not available in Newton. The reason for making this distinction is to not take credit for some of the intuitive solutions offered by Newton.

The evaluation is based on the framework presented earlier in this thesis, and the result is presented in Table 1. The first column contains the categories of the challenges; the second column the challenge, the third column a short description of the challenge and the last column contains the result from the evaluated solution. In the result column of Table 1 no mark means that the challenge was not addressed at all, one mark means partly addressed and two marks means that we think we have found a way to properly address the challenge. The evaluation is discussed more in detail in the following text.

6.1. Overview

Folder hierarchies on a file system allows for a somewhat logical grouping of information items as they can be placed in the same folder regardless of what application created them. Newton’s Content Map (Fig. 1) was designed to give the user the possibility to create relations between objects (not shown on picture) to better support a logical view of the information. Relations can be created almost freely between objects in the main window, for example a piece of information can have relations to several other objects and thus appear in many different contexts. Could better support of a logical view enhance the understanding of how different objects relate as well as decrease the time it takes to find information by browsing?

Newton’s Content Map presents the information in a tree structure, as it is based upon a folder hierarchy which is added by the user. As mentioned in the literature review, users inboxes are seldom more than two folders deep to allow fast access to items. Could this user behavior be translated to other contexts than the email inbox, since the principals are very similar? Newton addresses this by showing all the content of the selected node and it's sub-nodes in the Content List, thus allowing complex node structures but still keeping the overview. Creating an empty node on the map and populating the sub-nodes makes the empty node show an aggregated view of the information in the sub-nodes.

As the Content Map gets filled the scroll and zoom functions of Newton help the user keep an overview of the information.

As mentioned earlier a mere filename does not always seem to be enough to remind the user why the file was saved and its supposed use. As a remedy to this challenge the prototype offers the possibility for the user to add comments to files and nodes that are shown as tool tips when the mouse hover the item in the Content List.

6.2. Fragmentation

Information fragmentation is a big challenge for PIM to overcome as it affects all the areas presented in this thesis. One of the goals we had with creating the prototype was to allow for seamless interaction over the three information contexts, the local, the organizational and the global.

The RSS feed and Google Desktop nodes that we added to the prototype shows that content from multiple sources can be added, thus covering all three information contexts.

Files of any format can be imported into Newton, as long as it resides on a file system where the operating system Newton is running on can read it. Each file is still handled by its associated software but represented as content in Newton.

Fragmentation can also have a negative impact on the overview the user gets of related information in a project. Newton addresses this by searching for and displaying all instances of a file.

Information fragmentation arises because different related information items are situated in different information contexts and/or handled by different software. This makes it hard for the user to get a good overview of the available information related to, for example, a certain project. Newton does not eliminate this problem per se, but it helps the user locate all the occurrences of an information object.

6.3. Organization

The challenge of task management can be handled by using reminder notes in the prototype. There is a risk of the Content Map getting cluttered if used excessively however, and it cannot be seen as a replacement to a calendar. A possible solution could be that reminders would be added as content items,

Table 1. Result of how the prototype theoretically performs against our evaluation framework.

Overview	Relations	*associations between related objects*	++
	Logical view	*one item, multiple locations*	++
	Folder hierarchies	*"out of sight, out of mind"*	+
	Context	*descriptions*	+
Fragmentation	Multiple sources	*local, intranet, web*	++
Organization	Sorting and filtering	*"overloading"*	+
	Metadata and tagging	*content information*	+
	Task management	*reminders*	+
Searching	Keywords	*synonyms, homonyms*	
	Result list	*grouping*	+
	Cognitive load	*navigation over teleportation*	++

allowing the user to group reminders in a node, for example like a normal calendar by day, or perhaps as a sub-node to a project. The tasks would then also be sortable by date and other metadata.

The metadata supported in Newton are, for each file; title, comment, type and date. Future developments could include tagging of content items. The titles and comments combined could however be seen as a replacement to the normal tagging with keywords.

The content items can be sorted by the above mentioned metadata. As previously discussed it is common to re-find information by sorting by for example a date. A future addition to Newton could be support for drag and drop ordering of content items too, just like in the Content Map, to allow the user to order content items however she wants.

6.4. Searching

When searching in Newton the results are presented in the Content Map by the node(s) that they are located in, which is a kind of a grouping of results. The context to which each result is connected to is shown, allowing navigation rather than teleportation which eases the cognitive load on the user.

It is however only possible search by node and content item titles. A complete (full-text) index of all content added and more searchable metadata would allow the user to find more precise results than currently possible. The ability to add results from a Google Desktop query to a node solves this in a way. That search is however performed on content outside the application too, so an integrated index would certainly be more powerful.

7. DISCUSSION

As shown in the theoretical evaluation chapter of this thesis, the changes made to Newton in accordance to the framework were evaluated by simply running the prototype and decide whether or not the different solutions would get a score of “not addressed”, “partly addressed” or “fully addressed”. What this kind of evaluation fails to assert is the actual usability of the changes made. A fully fledged usability test would have been conducted if time were available as well as (maybe) distribution of the prototype for user testing and evaluation. This could help develop the newly implemented functionality as well as reveal the need for new functionality. Also we do not know if and what kind of user testing GlobeAccess has conducted on Newton, what results they got or how the design of it emerged. What we saw was a piece of software that had potential to do what we initially wanted, create seamless movement between the information contexts.

To access the global context (the Internet) the use of RSS feeds was introduced to the prototype. RSS feeds provide titles, descriptions and links to each item and are hence basically the same thing as a web bookmark. To access the organizational context (the intranet) you would also be able to connect nodes to RSS feeds, but a more usual location to find organizational information in are on a file server, which of course is supported as well. The local context was already covered in Newton, as you can import your current folder structures. We added the support to search the local context by allowing the user to connect a node to a Google Desktop search result.

In the overview section of the theoretical evaluation chapter the logical relations between information objects was discussed. This led us to wonder whether better support of a logical view of information could enhance the understanding of how different objects relate as well as decrease the time it takes to find information by browsing. As mentioned earlier, users would rather perform a task they recognize (browse) rather than a thinking task (keyword search) due to the cognitive load. Could more software that use folder structures, such as email clients, make use of a logical view to enable the user to faster find the right information or are there scenarios when tree structures are better suited for the task?

In [9] Jones et al. states that the users sent emails to themselves with details of the information item written in the mail. Would this behavior be eradicated with the use of searchable metadata tags, such as comments?

The original purpose for this thesis was to incorporate a search index from a platform such as Google Enterprise or Microsoft Search into Newton, by request of a company who cooperate with GlobeAccess. This focus shifted towards a general extension of Newton after the academic perspective of the thesis was decided (to create an evaluation framework for PIM software).

One of the main reasons for the small amount of functionality added in the prototype was because GlobeAccess did not want to give us access to the source code, something that was promised us early on in the project. Instead it was agreed that we would create design documents for the changes we wanted to do and hand these to the developer in charge of Newton and he would then incorporate the changes. When the time came to implement our ideas, GlobeAccess did not fulfill their part of our agreement and only implemented some of the new features.

8. CONCLUSION

By designing a framework for PIM tool evaluation and developing a prototype where we tried to solve some of the challenges we found, we believe that we have reached the goals with had with our research. 1) To identify and compile the challenges expressed in PIM literature. 2) To devise a prototype interface that addresses (some of) these challenges.

The evaluation framework we designed summarizes challenges in PIM that previously were scattered in many different articles. It presents features that a PIM tool of today should have. We were not able to perfect the prototype the way we wanted, but GlobeAccess, the company developing Newton, found our work useful and they will implement more of our ideas in the future. Other developers can also use the framework to evaluate their existing PIM tool, or as a starting point for a new tool.

REFERENCES

1. Jadaan, T. and Stenmark, D. (2008). "Knowledge workers' use of electronic information sources", in *Proceedings of ECIS 2008*, Galway, Ireland, June 9-11, 2008.
2. Jones, W. Phuwanatnurak, A. J. Gill, R. Bruce, H. (2005) " Don't take my folders away!: organizing personal information to get things done" in *CHI '05 extended abstracts on Human factors in computing systems*, Portland, OR, USA, pp. 1505 – 1508.
3. Whittaker, S. and Sidner, C. (1996) "Email overload: exploring personal information management of email" in *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, Vancouver, British Columbia, Canada, pp. 276-283.
4. Ducheneaut, N. and Bellotti, V. (2001). "E-mail as habitat: an exploration of embedded personal information management", *Interactions*, 8(5), Sep. 2001, pp. 30-38.
5. Fisher, D. Brush, A. J. Gleave, E. Smith, M. A. (2006) "Revisiting Whittaker & Sidner's "email overload" ten years later" in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, Banff, Alberta, Canada, pp. 309 – 312.
6. Cockburn, A., Greenberg, S., McKenzie, B., Jasonsmith, M., and Kaasten, S. (1999). WebView: A Graphical Aid for Revisiting Web Pages. in *Proceedings of the OZCHI'99 Australian Conference on Human Computer Interaction*, (November 28-30, Wagga Wagga Australia).

7. Dumais, S. Cutrell, E. Cadiz, J. J. Jancke, G. Sarin, R. Robbins, D. C. (2003) “Stuff I've seen: a system for personal information retrieval and re-use” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval,* Toronto, Canada, pp. 72-79.
8. Jones, W. Bruce, H. Dumais, S (2001) “Keeping found things found on the web” in *Proceedings of the tenth international conference on Information and knowledge management*, Atlanta, Georgia, USA, pp. 119-126.
9. Jones, W. Bruce, H. Dumais, S (2005). “Once found, what then? A study of “keeping” behaviors in the personal use of Web information” in *Proceedings of the American Society for Information Science and Technology*, pp. 391-402.
10. Dong, X. Halevy, A. (2005) “A platform for personal information management and integration”, CIDR.
11. Jones, W. Klasnja, P. Civan, A. Adock, M.L. (2008) ”The Personal Project Planner: Planning to Organize Personal Information” in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems,* Florence, Italy, pp. 681-684.
12. Boardman, R. and Sasse M.A. (2004). “Stuff Goes into the Computer and Doesn’t Come Out. A Cross-tool Study of Personal Information Management” in *Proceedings of CHI 2004*, (April 24-29, 2004)
13. V. Bush. As we may think. *The Atlantic Monthly*, July 1945.
14. Stenmark, D. and Jadaan, T. (2008). "Email as an Integration Device: A Study of Work Place Information Seeking". *in Proceedings of AMCIS 2008* Toronto, Canada, August 14-17, 2008.
15. Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. (2004) The perfect search engine is not enough: a study of orienteering behavior in directed search, in *Proceedings of CHI ’04*, April 24-29, Vienna, Austria, 415-422.
16. Dumais, S. Cutrell, E. & Chen, H. (2001). “Optimizing search by showing results in context” in P*roceedings of CHI ‘01,* Seattle, WA., pp. 277-284.

APPENDIX A

Elsweiler, D. "Supporting human memory in personal information management". ACM New York, NY, USA. 2008.

Barreau, D. and Capra, R. and Dumais, S. and Jones, W. and Pérez-Quiñones, M. "Introduction to keeping, refinding and sharing personal information". ACM New York, NY, USA. 2008.

Bernstein, M. and Van Kleek, M. and Karger, D. and Schraefel, MC. "Information scraps: How and why information eludes our personal information management tools". ACM New York, NY, USA. 2008.

Bergman, O. and Beyth-Marom, R. and Nachmias, R. and Gradovitch, N. and Whittaker, S. "Improved search engines and navigation preference in personal information management". ACM New York, NY, USA. 2008.

Karger, D. and Bernstein, M. and Van Kleek, M. and Khushraj, D. and Nayak, R. and Liu, C. and Karger, D.R. "Wicked Problems and Gnarly Results: Reflecting on Design and Evaluation Methods for Idiosyncratic Personal Information Management Tasks". 2008.

Elsweiler, D. and Baillie, M. and Ruthven, I. "Exploring memory in email refinding". ACM New York, NY, USA. 2008.

Jones, W. and Klasnja, P. and Civan, A. and Adcock, M.L. "The personal project planner: Planning to organize personal information". ACM New York, NY, USA. 2008.

Jain, J. and Ghosh, R. and Dekhil, M. "Multimodal capture of consumer intent in retail". ACM New York, NY, USA. 2008.

Teevan, J. "How people recall, recognize, and reuse search results". ACM New York, NY, USA. 2008.

Teevan, J. and Jones, W. and Capra, R. "Personal information management (PIM) 2008". ACM New York, NY, USA. 2008.

Panovich, K. and Vargas, G. and Karger, D. "Note to Self: Examining Personal Information Keeping in a Lightweight Note-Taking Tool". 2009.

Meyers, E.M. and Fisher, K.E. and Marcoux, E. "Making Sense of an Information World: The Everyday-Life Information Behavior of Preteens". UChicago Press. The Library Quarterly vol 79 no 3 pp 301-431. 2009.

Komlodi, A. and Lutters, W.G. "Collaborative use of individual search histories". Elsevier. Interacting with Computers vol 20 no 1 pp 184-198. 2008.

Barreau, D. "The persistence of behavior and form in the organization of personal information". Wiley Subscription Services. Journal of the American Society for Information Science and Technology vol 59 no 2. 2008.

Adar, E. and Teevan, J. and Dumais, S.T. "Large scale analysis of Web revisitation patterns". ACM New York, NY, USA. 2008.

Kay, J. "Lifelong Learner Modeling for Lifelong Personalized Pervasive Learning". IEEE. IEEE Transactions on Learning Technologies vol 1 no 4 pp 215-228. 2008.

Ma, S. and Wiedenbeck, S. "File management with hierarchical folders and tags" in "Proceedings of the 27th international conference extended abstracts on Human factors in computing systems" pp 3745-3750. ACM New York, NY, USA. 2009.

Kettler, B. and Starz, J. and Hingst, R. and Losco, J. and Hoffman, M. "Putting knowledge in context to facilitate collaboration" in "Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on" pp 313-320. 2008.

Sauermann, L. and Heim, D. "Evaluating long-term use of the gnowsis semantic desktop for pim" in "Proceedings of the 7th International Conference on The Semantic Web" pp 467-482. Springer. 2008.

Van Kleek, M.G. and Panovich, K. and Vargas, G.G. and Karger, D.R. and Schraefel, MC. "Note to self: examining personal information keeping in a lightweight note-taking tool" in "Proceedings of the 27th international conference on Human factors in computing systems" pp 1477-1480. ACM New York, NY, USA. 2009.

Teevan, J. and Cutrell, E. and Fisher, D. and Drucker, S.M. and Ramos, G. and André, P. and Hu, C. "Visual snippets: summarizing web pages for search and revisitation" in "Proceedings of the 27th international conference on Human factors in computing systems" pp 2023-2032. ACM New York, NY, USA. 2009.

Adar, E. and Teevan, J. and Dumais, S.T. "Resonance on the web: web dynamics and revisitation patterns" in "Proceedings of the 27th international conference on Human factors in computing systems" pp 1381-1390. ACM New York, NY, USA. 2009.

Miller, R.C. and Chou, V.H. and Bernstein, M. and Little, G. and Van Kleek, M. and Karger, D. "Inky: a sloppy command line for the web with rich visual feedback" in "Proceedings of the 21st annual ACM symposium on User interface software and technology" pp 131-140. ACM New York, NY, USA. 2008.

Kalnikaite, V. and Whittaker, S. "Cueing digital memory: how and why do digital notes help us remember?" in "Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008" pp 153-161. British Computer Society Swinton, UK, UK. 2008.

Ryall, J. "Reminding and refinding: examining how software developers use annotations". University of Victoria. 2008.