

Americas Conference on Information Systems (AMCIS)

AMCIS 2010 Proceedings

Association for Information Systems

Year 2010

Design Implications for Personal
Information Management: A Theoretical
Evaluation of a Prototype Interface

Dick Stenmark*

Klas Espenkrona[†]

Mattias Svensson[‡]

*Department of Applied IT Gothenburg University, Sweden, dicks@ituniv.se

[†]Department of Applied IT, Gothenburg University, Sweden, klas.espenkrona@gmail.com

[‡]Department of Applied IT Gothenburg University, Sweden, mattias.swenson@gmail.com

This paper is posted at AIS Electronic Library (AISeL).

<http://aisel.aisnet.org/amcis2010/85>

Design Implications for Personal Information Management: A Theoretical Evaluation of a Prototype Interface

Dick Stenmark

Department of Applied IT
Gothenburg University, Sweden
dicks@ituniv.se

Klas Espenkrona

Department of Applied IT
Gothenburg University, Sweden
klas.espenkrona@gmail.com

Mattias Svensson

Department of Applied IT
Gothenburg University, Sweden
mattias.swenson@gmail.com

ABSTRACT

Personal Information management (PIM) is a research area that receives interest from a variety of disciplines including human-computer interaction, information retrieval, information systems research and psychology to mention but a few. The diversity in approaches and the cross-disciplinary nature of PIM have resulted in a fragmented picture of the problems and challenges designers of PIM tools are facing. In this paper we present a PIM evaluation framework based on a broad literature study of the known challenges within PIM. Focusing in particular on information fragmentation and the re-finding of information, we built and evaluated a PIM prototype using our framework. We found that zooming, separation between logical and physical structures, and showing search results in context seem like useful future design ideas.

Keywords

Personal Information Management, PIM, evaluation, design.

INTRODUCTION

Personal Information Management (PIM) is an umbrella term covering the everyday production, acquiring, organising, maintaining, storing and retrieval of information that is not primarily *private* but *personal* (Barreau, 1995). A substantial body of research on various aspects of PIM in many different settings has begun to pile up in recent years (cf. Jones & Teevan, 2007). One particularly pervasive PIM problem is that of information fragmentation (Karger, 2007), i.e., the fact that information related to a single task is often scattered across several different applications and environments. A typical example would be project information, where the project specification may be in a Word document, the budget in an Excel file, communication with the customer and the project manager may be kept in emails, and other resources may reside on the intranet or on the Web. A project member may have seen all these documents but may later have trouble re-accessing them. Karger (2007) argues that grouping of related information is a central PIM activity currently hindered by the artificial separation imposed by the different applications.

Looking at how office workers find their information, Jadaan & Stenmark (2008) suggested three high-level information environments between which the users need to move; the local, the organisational, and the global. A central problem is that today's information seeking tools typically are designed for and constrained to one of these environments at a time. Towards a solution to the problem of re-finding and re-accessing information spread across all these three environments, it has been argued that a unifying device that would span many different sources or that would integrate between different (PIM) tools would simplify the users' situation (Boardman & Sasse, 2004; Dumais et al., 2003; Jones et al., 2002).

Three major categories of unifying approaches have been identified: The use of metadata, the use of standardised data types, and visual unification (Karger, 2007). In this paper, we have studied visual unification. Boardman and Sasse (2004) note that two types of approaches exist when trying to improve integration between tools. The first approach is embedding support in an existing tool, i.e. in an email client (Ducheneaut & Bellotti, 2001; Whittaker & Sidner, 1996). The second is to provide a new, unifying, interface (Cockburn et al., 1999; Dumais et al., 2003). Our focus is on the latter. The most obvious form of visual unification is the one provided by the operating system's windows manager. The windows manager handles the opening, resizing, moving, stacking, and closing of these windows, allowing the user to have open many windows simultaneously and manipulate all sorts of information objects. Although familiar, this solution is far from optimal. Firstly, the application needs a rather large window because not only the information object needs to be visualised, the application also has menus and toolbars that need to be displayed. This limits the number of windows that can be displayed at the same time. Secondly, the window manager treats the application as a rectangle of pixels and do not allow the user to select out only

the sub-portion that is of interests. These limitations result in a desktop filled with windows obscuring one another (Karger, 2007).

In this paper we are thus focusing on the problem of re-finding (project) information. Based on a literature study of currently identified PIM challenges, we compiled a framework for PIM tool evaluations and built an experimental prototype application that provides a unifying interface on top of existing tools. Evaluating the prototype against the framework led to new insights about unification. Next, we describe our research approach in more detail. We then introduce our framework and the prototype in the two following sections. Thereafter we carry out a theoretical evaluation before ending the paper with our conclusions.

RESEARCH METHOD

Our work entails two parts: Firstly, a thorough literature review in order to identify the PIM challenges acknowledge by the research community, and; Secondly, the development of a prototype application to resolve (some of) these challenges.

Even if the phrase PIM was first mentioned in the 1980s, the problem itself is much older and obviously there is a vast amount of literature to choose amongst. We first engaged Google Scholar to search for the phrase “personal information management”, which resulted in close to 6,000 hits. Out of these, we selected and analysed the ten most cited papers. We then clicked on “new articles” and asked for paper from 2008 or newer. This produced 558 hits. We therefore narrowed the query by adding the term “refind” and ended up with 25 papers. These were also included in our analysis. Using these 35 papers, and the recent edited book on PIM (Jones and Teevan, 2007), as a base, we also followed references to both newer and older papers that appeared to address the problems of unification and re-finding. While reading, all challenges and problems mentioned in the papers were noted in a spreadsheet and iteratively reorganised 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 result of this process is presented in the Framework for evaluation section below.

The prototype was built as an add-on to an already existing product called Newton. Together with the company that develops Newton, we came to an agreement that we were to provide design ideas to their chief engineer, who then would incorporate these into a test version of the product and return it to us for evaluation, which would generate new ideas. This iterative process went on for approximately four weeks. In its native form, Newton provided a graphical and zoomable map interface to local desktop content such as files and documents. In line with our theoretically informed agenda, we extended this to cover also the organisational and the global information environments by adding support for networked files, intranet pages, web bookmarks, RSS feeds and search queries from desktop, intranet or Web search. More details are provided in the section entitled The Prototype.

Evaluating PIM tools is an important but non-trivial task that presents several methodological challenges. Kelly and Teevan (2007) describe four evaluation approaches - naturalistic, longitudinal, case study, and laboratory - but conclude that they all are limited in different aspects. There are three major challenges: (1) incorporation of realistic PIM situations for test purposes is difficult; (2) PIM actions are sporadic and unpredictable and thus difficult to capture, and; (3) new PIM tools disrupt users' ordinary routines and thus cause problems even if the features are good (Kelly & Teevan, 2007). In this paper, we therefore suggest and use a fifth approach when carrying out a theoretical evaluation using an evaluation framework.

FRAMEWORK FOR EVALUATION

Our literature review revealed four major challenges: fragmentation, organisation, overview and searching. These topics obviously partly overlap but having analysed the problems addressed in previous research we found information overview to be the main issue, and fragmentation and organisation more specialised aspects of the overview challenge. Searching, although meant to be a solution, contains its own challenges. We illustrate this in Figure 1.

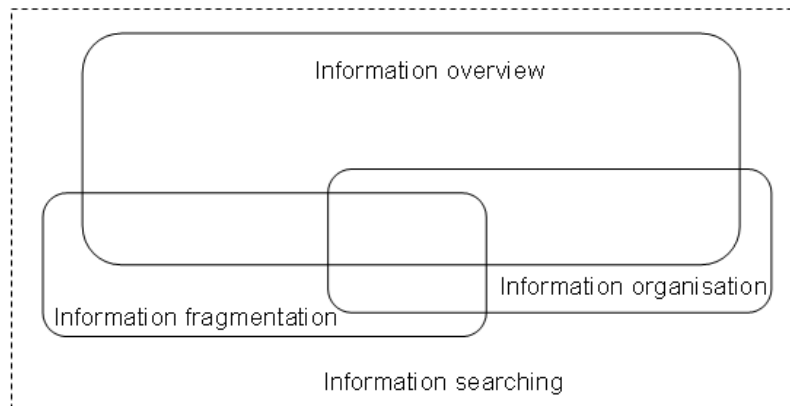


Figure 1. Four challenging areas of PIM.

Information overview

The overarching challenge seems to be to get a useful overview of one's information items. In a strict folder hierarchy, items have unique placements and cannot appear in multiple places even though they may have characteristics that could be used to group them according to multiple schemas (Jones et al., 2005). However, once information is placed in a folder it becomes out of sight and thus also "out of mind" (Jones et al., 2005). This stems from the fact that folders are opaque and do not disclose their content until explicitly commanded to do so. This fact severely hampers a good overview.

The information technology that supports PIM at the same time adds to the overview problem by enabling access to too much information, thereby letting less important stuff hide what is more important (Teevan et al., 2006). The large number of information objects kept has been identified as a problem for folders (Jones et al., 2002; 2005) as well as for email (Whittaker & Sidner, 1996). Unread emails, for example, are often highlighted to attract attention but emails kept as a reminder of an upcoming deadline do not call out for attention. A challenge for PIM tools is thus to increase awareness without causing distraction (Cutrell et al., 2006).

Compared to searching in general, finding personal information more often relies on contextual knowledge from where the information was previously encountered (Cutrell et al., 2006). However, many (PIM) tools provide too little context to facilitate re-finding once the information has been saved. The typical example is the Bookmark or Favourite feature found in web browsers. Context information can be provided in the link title or in the folder name, but still this provides very limited context. Instead, many users email URLs to themselves so that they can add their own annotations (Jones et al., 2002). One way to address this is tagging. Jones et al. (2008) note that support for tagging that is format independent is still lacking, and more PIM research in this area is thus needed.

The challenges of fragmentation and organisation further add to the challenge of getting a useful overview, but also contain their specific problems.

Information organisation

Dong and Halevy (2005) remind us that humans do not think by way of hierarchies but by way of associations between objects. A big challenge for PIM is thus that our tools with few exceptions are built around (file) hierarchies when what we need are tools to allow browsing by association in a *logical* rather than *physical* structure (Dong & Halevy, 2005). Folders cannot easily be rearranged so that information currently used in e.g. a project is kept in close proximity, and users therefore rename folders to start with "a" or "aa" to force them to the top. Support for folder restructuring has been highlighted as a promising area for future research but currently such features are rare (Jones et al., 2005).

Personal information is often kept and organised as a reminder of things to do. A folder hierarchy may also function as such a reminder or a to-do list (Jones et al., 2002; 2005). Another often used approach is to email to-do list to one self (Ducheneaut & Bellotti, 2001) or to simply keep email in the inbox to signal that further actions are required (Whittaker & Sidner, 1996). However, our desktop and our applications only offer limited space where these reminders can be placed and once the

amount of items has grown large enough the “reminders” will hide each other. A PIM challenge is thus to provide reminders and awareness cues to the users.

Information fragmentation

Information fragmentation is often referred to as a challenge for PIM (Bergman et al., 2006; Jones et al., 2008; Karger, 2007; Marshall & Jones, 2006). The problem of information fragmentation can be defined as the fact that the information needed to complete a task may be scattered across different applications and formats and the compilation of the needed pieces may thus be time-consuming at best and generate errors at worst (Jones et al., 2008).

Several applications are used many times a day in order to re-find pieces of information needed to compile a report or presentation. On a higher level, users need to shift between three information environments; the local, containing saved emails and hard disk files; the organisational, holding intranet sites and shared file servers; the global, with Web resources (Jadaan & Stenmark, 2008). All these items must be located and accessed using different applications.

In addition, information can be kept in many different formats and for many different reasons. The same piece of information may be kept in multiple formats to increase the likelihood that it later will be re-found. Sometimes an information object is simply copied to another storage location because the user wants to view it together with other related objects, e.g. project information (Bergman et al., 2006). The down-side of these data duplication efforts are that they add to the fragmentation and make it difficult to maintain the integrity of the information if it needs to be updated (Marshall & Jones, 2006).

Information search

Searching for electronic information often means engaging some sort of information retrieval tool such as a web search engine. One aspect of searching that clearly relates to information overview is the presentation of the search results. Search results, regardless of what tool produced them, are typically always presented in a list format. This is unfortunate since grouping of results, which enhances overview, enables users to quicker find what they were looking for. Clustering interfaces are shown to be more effective even when lists are augmented with category names for each result (Dumais et al., 2001). Desktop search share much of the same limitation even if it has features that Web search engines lack, such as sorting on different attributes or restringing output to only emails or only documents (Russell & Lawrence, 2007).

Another problem not directly linked to information overview is the common reliance on keyword-based searching. The most fundamental shortcoming of this method is the fact that it is a recall-based method (Good et al., 2005). Ordinary users want to achieve their goals with a minimum of cognitive load and therefore prefer *recognition* tasks to *recall* tasks. This means that they rather browse to information than use keyword search even if the former takes longer, since most people are willing trade time to minimise complexity (Marchionini, 1992). PIM tools should thus support recognition rather than recall.

Keyword search is problematic also because a user’s vocabulary often does not match the desired document’s vocabulary. This means that users may not be able to find certain documents or they may have to experiment with several queries before they find what they want (Good et al., 2005). Keyword search has problem distinguishing between words that are spelled identically but have different meanings (homonyms) and cannot find documents that contain words that have the intended meaning but are spelled differently (synonyms). The former presents a precision problem whereas the latter presents a recall problem (Svenonius, 1990). Search methods not relying entirely on keyword-based retrieval should thus be researched.

THE PROTOTYPE

The prototype was built as an add-on to an already existing product called Newton. 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 organisational, 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 organisation or to an RSS feed. This way, Newton can provide users with the ability to span the local, the organisational, and the global information context via a single interface.

Newton offers a graphical map-based interface to data, which resembles that of a mindmap (see Figure 2). 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 organised 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 the index 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.

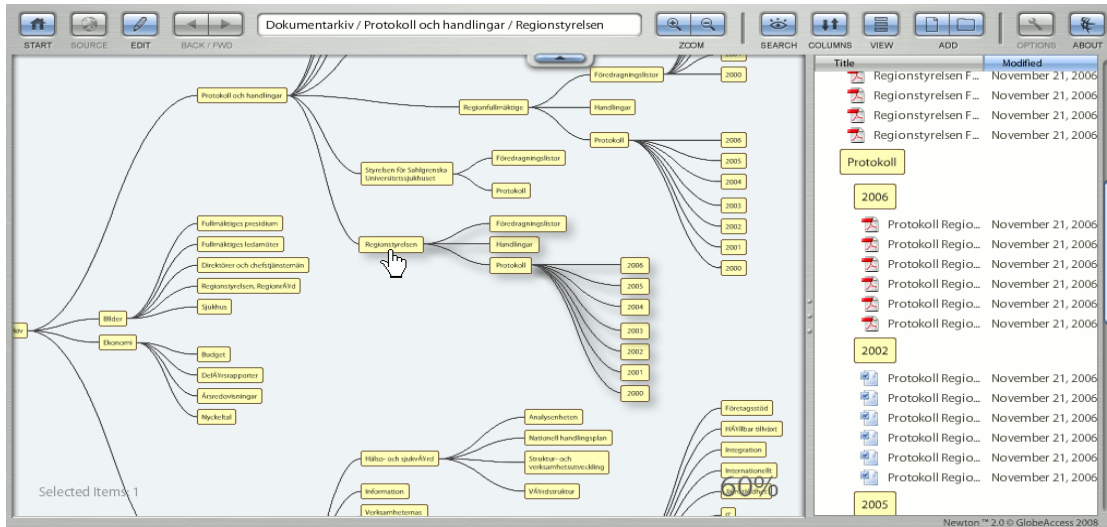


Figure 2. Clicking a node in the Content Map section opens the Content List on the right hand side.

The second part of the graphical user interface is called the Content List since it contains a detail list of all the individual items contained in a node. 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.

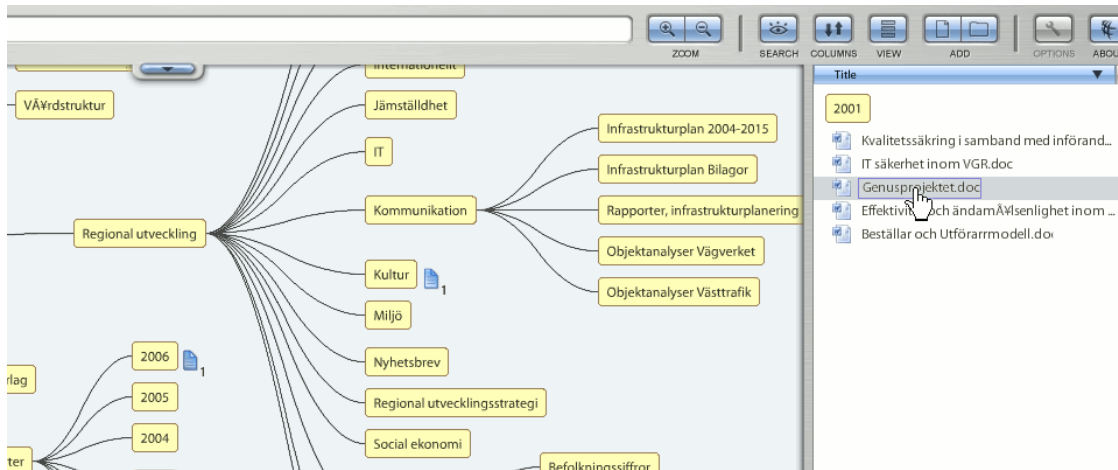


Figure 3. Selecting an information object in the Content List shows all instances of that object by highlighting the attention icon in the Content Map

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, sortable on name, type or date. 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 (see Figure 3). This method can also be used to find objects based on file type or modification date, which are the attributes currently supported in the prototype. The titles of the nodes and the

content objects 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.

THEORETICAL EVALUATION

Information overview

The zoom and scroll features that Newton offers allow the user to get both an overview and a detailed view from within the same tool, thus not needing to open multiple windows. Since the tree is always fully expanded (as opposed to file structures which is often collapsed to show only one level), Newton provides a good overview whilst yet allowing the user effortlessly and seamlessly to zoom in to a close-up of any sub-section of the tree. This feature partly addresses the problem of scale. Traditional hierarchies typically grow out of the screen rather quickly if expanded and obscure information when collapsed. Studies of email use have shown that hierarchies seldom are more than two levels deep (to allow fast access to items) and that folder nesting often was a response to limited screen space (Ducheneaut & Bellotti, 2001). We argue that a zoomable mapping allows for a deeper (i.e., more detailed) structure without risking losing sight of information or running out of screen and this should be advantageous.

Whilst the traditional file hierarchy presents the tree structure in a small window and the content of a certain folder in the main window, Newton does the opposite. The Content Map, which shows the overall structure, occupies the main portion of the screen whilst the Content List exposes the individual information objects in a smaller screen as seen in Figures 2 and 3. This, we argue, adds more context and enhances the users understanding of how different objects relate as well as decreases the time it takes to browse to information. This appears to be a useful design solution in future PIM applications.

Information organisation

We find the separation of the physical structure and the logical view to be a very promising design implication. Since the view is based on a logical and not physical structure, (references to) objects can freely be copied between nodes without risking inconsistencies when one version is updated. Having multiple physical copies is otherwise a very common way of trying to bring all information relevant to a certain task or project together.

Newton's content map allows the visualisation of not only the information objects but also their relationships. New nodes can be created and existing nodes can be moved and linked to existing nodes to indicate sub-tasks without having to relapse into "aa" renaming or similar workarounds. Relations can be created almost freely between objects in the main window, e.g. an information object can have relations to several other objects and thus appear in many different contexts. This setup enables the user to organise his or her information objects according to ongoing tasks and personal preferences. This is also a promising design idea.

Another task management feature in Newton is the reminder notes that the system provides. Attached to information objects, these reminders can attract the user's attention and point to unfinished tasks. This solution is obviously less than perfect since there is a risk of the content map getting cluttered if notes are used too excessively. A possible future solution could be to implement these reminders as information objects in their own right, allowing the user to collect reminders in a node, for example like a normal calendar by day, or perhaps as a sub-node to a project.

Information fragmentation

To address the fragmentation problem, Newton enables seamless interaction over all three information contexts; the local, the organisational and the global. The local information environment is covered by the default features of Newton that displays (a selected portion of) the content of the file system, and the added support for stored emails. The organisational information environments is covered by function calls to the intranet search engine, and the inclusion of network drives, and the global information environments is covered by a call to a public Web search engine. Displaying information from all these contexts in the same view lessens the problem of fragmentation and is thus a useful design implication.

Fragmentation also arises when the user is forced to duplicate information items. Although our prototype design does not eliminate this need *per se*, it helps the user find all occurrences of a particular information object via the attention icon shown in Figure 3. This also seems to be a useful design idea for future PIM tools.

Information search

Searching can be done by browsing or by invoking a search tool. Newton's strength lies more in the browsing than in the search tool approach and since recognition is less mentally taxing than recall, this is arguably a positive feature. There is still

a search function in Newton and although it only searches object names and metadata, it provides a rather interesting feature. When searching in Newton, the results are presented both in the Content map and in the Content list. The notification icon is lit by each node in the Content map that contains a hit, indicating the number and the location of the hits. This provides a grouping and a context that we believe the user would benefit from. A full-text index of all content and support for more searchable metadata would obviously enhance search further, but though the ability to import results from a Google Desktop query to a node partly solves this, the problems with keywords-based search still remains.

However, our prototype only searched information that the user had imported, and for newly arrived emails and updated files and/or web pages to be searchable, the corresponding nodes must be manually refreshed. Here is room for improvements since our prototype is far from perfected, but we find that the ability to show in context (i.e., on the map) where search results are located is a useful feature.

CONCLUSIONS

There were two goals with our research: 1) To identify and compile a set of challenges expressed in PIM research literature, and; 2) To devise a prototype interface that addresses (some of) these challenges in order to derive new design implications. This paper provides a theoretical framework that we believe can be used also by other researchers wanting to test and evaluate PIM applications. Although our framework may not be exhaustive, it summarises many PIM challenges that previously were scattered across many different articles, and it should serve as a useful base for further development and thus constitute a contribution. A summary of our findings are found in Table 1.

Problem area	Challenge	Examples	Score	Comment
Overview	Opaqueness	<i>"out of sight, out of mind"</i>	+	Support
	Amount of info	<i>obscuring important stuff</i>	+	
	Lack of context	<i>richer descriptions</i>	+	
Organisation	Associations	<i>information relations</i>	++	Strong support
	Re-arrangements	<i>folder restructuring</i>	++	
	Task management	<i>Reminders, to-do lists</i>		
	Logical view	<i>arrange according to task</i>	++	
Fragmentation	Multiple sources	<i>local, organisational, global</i>	++	Support
	Multiple formats	<i>inconsistencies</i>	+	
	Multiple copies	<i>inconsistencies</i>	+	
Searching	Cognitive load	<i>recognition vs. recall</i>	++	Support
	Keywords	<i>synonyms, homonyms, vocabulary</i>		
	Result list	<i>grouping</i>	+	

Table 1. Summarising our theoretical evaluation

The prototype was developed in parallel to the compilation of the framework and served as a method to test the framework as well as test new design ideas. Although we have argued that user tests of PIM tools tend to be artificial and thus provide biased and unreliable results, an extension of this work would be to collect experiences from people using tools that exploit some of the features this study has identifies as promising. Users are known to be unpredictable in their preferences and behaviour, and our theoretical evaluation is thus somewhat limited. Still, the development of the prototype and the evaluation framework has enabled us to identify a set of generally applicable design ideas that should be considered in future PIM tools:

- A zooming ability can address overview, fragmentation and search challenges.
- Separate the logical view of the information from the physical structure.
- Provide a large Content map and a smaller Content list rather than the other way around.
- Support the ability to show in context (i.e., on the map) where search results are located.

REFERENCES

1. Barreau, D.K., (1995) Context as a factor in personal information management systems, *Journal of the American Society for Information Science*, 46(5), 327-339.

2. Bergman, O., Beyth-Marom, R., and Nachmias, R. (2006) The project fragmentation problem in personal information management, in *Proceedings of CHI '06*, Montréal, Canada, 271-274.
3. 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 '04*, Vienna, Austria, 583-590.
4. 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*, Wagga Wagga, Australia.
5. Cutrell, E., Dumais, S. T., and Teevan, J. (2006) Searching to eliminate personal information management, *Communication of the ACM*, 49(1), 58-64.
6. Ducheneaut, N. and Bellotti, V. (2001) E-mail as habitat: an exploration of embedded personal information management, *Interactions*, 8(5), 30-38.
7. Dumais, S., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R. and Robbins, D.C. (2003) Stuff I've seen: a system for personal information retrieval and re-use, in *Proceedings of the SIGIR '03*, Toronto, Canada, 72-79.
8. Dumais, S., Cutrell, E., and Chen, H. (2001) Optimizing search by showing results in context, in *Proceedings of CHI '01*, Seattle, WA. 277-284.
9. Dong, X. and Halevy, A. (2005) A platform for personal information management and integration, in *Proceedings of CIDR '05*, Asilomar, CA.
10. Good, L.E., Popot, A. C., Janssen, W. C., and Bier, E. A., (2005) Fluid Interface for Personal Digital Libraries, in *Proceedings of ECDL 2005*, Vienna, Austria, pp. 162-173.
11. Jadaan, T. and Stenmark, D. (2008) Knowledge workers' use of electronic information sources, in *Proceedings of ECIS 2008*, Galway, Ireland.
12. Jones, W. Bruce, H. Dumais, S. (2002) Once found, what then? A study of "keeping" behaviors in the personal use of Web information, in *Proceedings of the ASIST 2002*, 391-402.
13. Jones, W. Klasnja, P. Civan, A. Adock, M.L. (2008) The Personal Project Planner: Planning to Organize Personal Information, in *Proceeding of CHI 2008*, Florence, 681-684.
14. Jones, W. Phuwatnurak, A. J. Gill, R. Bruce, H. (2005) Don't take my folders away!: organizing personal information to get things done, in *Extended abstracts of CHI '05*, Portland, OR, 1505-1508.
15. Jones, W. and Teevan J. (2007) Personal Information Management, University of Washington Press, Seattle, WA.
16. Karger, D.R. (2007) Unify everything: It's all the same to me, in (Jones & Teevan, eds.) Personal Information Management, University of Washington Press, Seattle, WA, 127-152.
17. Kelly, D. and Teevan, J. (2007) Understanding what works: Evaluating PIM tools, in (Jones & Teevan, eds.) Personal Information Management, University of Washington press, Seattle, WA., 190-204.
18. Marchionini, G. (1992) Interfaces for end-user information seeking, *Journal of the American Society for Information Science*, 43(2), 156-163
19. Marshall, C. C. and Jones, W. (2006) Keeping encountered information, *Communication of the ACM*, 49(1), 66-67.
20. Russell, D. and Lawrence, S. (2007) Search everything, in (Jones & Teevan, eds.) Personal information management, University of Washington Press, Seattle, WA., 153-166.
21. Svenonius, E. (1990) Design of Controlled Vocabularies, in (Kent, ed.) Encyclopedia of Library and Information Science, Vol. 45, Marcel Dekker, Inc., New York, NY., 82-109.
22. 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 the CHI '04*, Vienna, Austria, 415-422.
23. Teevan, J., Jones, W., and Bederson, B.B. (2006) Personal Information Management, *Communications of the ACM*, 49(1), 40-43.
24. Whittaker, S. and Sidner, C. (1996) Email overload: exploring personal information management of email, in *Proceedings of CHI '96*, Vancouver, Canada, 276-283.