The Associative PDA

An Organic User Interface for Mobile Personal Information Management

Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Jonathan Diehl

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Thomas Seidl

Registration date: April 15th, 2006
Submission date: Nov 15th, 2006

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, November 15th, 2006

Contents

List of Figures

List of Tables

Abstract

This thesis describes the design of the Associative PDA, a mobile user interface for personal information management, that is the organization of all the information relevant on a daily basis. It is based on the associative network, organizing information by directly connecting related items with each other. To find information, the network is traversed or a search is performed in the shared neighborhoods of relevant items.

Before starting the design process, we have conducted a contextual inquiry to understand the current practices of PIM. I have observed ten users in the field while doing their work to learn from their problems and strategies. The findings of this study have led to many interesting insights into the challenges of PIM.

We have developed the Associative PDA in three cycles of an iterative, user-centered design process. First, we have created storyboards to illustrate our design ideas and evaluated their feasibility with the users. Then, we have created a paper prototype to test the interface and gather early feedback about the basic interactions and visualizations of the system. Finally, we have designed a prototype in Adobe Flash to show its utility and collect feedback about details of the interface.

The final prototype was also used for an additional evaluation where we have compared our system with a commercially available reference system. This study has pointed out a general problem of evaluating PIM systems.

The outcome of this project is a working, interactive prototype and a list of issues that should be addressed to further improve the system.

Überblick

Diese These beschreibt den Assoziativen PDA, eine mobile Benutzerschnittstelle für persönliches Informationsmanagement (PIM), wobei es sich um die Organisation aller Informationen des täglichen Gebrauchs handelt. Das System basiert auf dem Assoziativen Netzwerk, worin Informationen organisiert werden indem sie direkt miteinander verbunden werden. Um Informationen aufzufinden kann das Netzwerk durchlaufen werden oder es können die Nachbarschaften mehrerer relevanten Informationsobjekte durchsucht werden.

Vor dem Beginn des Designprozesses haben wir eine kontextabhängige Untersuchung durchgeführt, um die Praxis des PIM kennen zu lernen. Ich habe zehn Benutzer im Feld während ihrer Arbeit untersucht, um von ihren Problemen und Strategien zu lernen. Der Befund dieser Studie hat viele interessanten Einsichten in die Schwierigkeiten von PIM erbracht.

Wir haben den Assoziativen PDA in drei Zyklen eines iterativen, benutzerzentrierten Design Prozesses entwickelt. Zuerst haben wir Storyboards erstellen, um unsere Design Ideen zu illustrieren und deren Machbarkeit auszuwerten. Dann haben wir einen Papierprototypen zum Testen der Benutzeroberfläche und zum frühen Sammeln von Meinungen über Interaktionen und Visualisierungen erstellt. Zuletzt haben wir einen Prototypen in Adobe Flash entwickelt, um die Nützlichkeit des Systems zu zeigen und Feedback über Einzelheiten der Benutzeroberfläche zu sammeln.

Der letzte Prototyp wurde ferner für eine zusätzliche

Studie genutzt, um unser System mit einem kommerziellen Referenz-System zu vergleichen. Durch diese Studie wurden wir auf ein generelles Problem an der Auswertung von PIM System aufmerksam gemacht.

Das Ergebnis dieses Projektes ist ein funktionierender, interaktiver Prototyp und eine Liste von Problemen und Ideen, die angegangen werden söllten um das System in Zukunft weiter zu verbessern.

Acknowledgements

Several people have worked together with me on this project. I want to thank Jan Borchers, for providing excellent guidance throughout the entire project and the numerous contributions he has made to the design. I also want to thank David Holman for the many hours he has invested, reviewing and supporting my work on the project, and the various ideas he has contributed to the design. Finally, I want to thank Thorsten Karrer for hard work he has done laying the groundwork for this project and all the fruitful discussions.

In addition, I want to thank Tico Ballagas and Monal Chokshi for their great support in various matters and acutely interesting conversations about my work.

Special thanks also go to Wolfgang Schmitz for reviewing the entire document on such a short notice.

Finally, I want to thank all the participants of the evaluations I have conducted: The patient workers I have observed during the contextual inquiry and the inspiring users I have tested my prototypes with. Without your feedback and suggestions this project could not have reached its current level of maturity.

Conventions

Throughout this thesis, I use the following conventions:

The whole thesis is written in American English.

There are two central characters in this thesis: the user and the designer. Instead of avoiding the singular pronoun, I decided to use 'she', whenever I am referring to the user, and 'he' whenever I am referring to the designer. This is not to imply anything about the composition of the actual population.

Chapter 1

Introduction

"Here's to the crazy ones, the misfits, the rebels, the troublemakers, the round pegs in the square holes... the ones who see things differently - they're not fond of rules... You can quote them, disagree with them, glorify or vilify them, but the only thing you can't do is ignore them because they change things... they push the human race forward, and while some may see them as the crazy ones, we see genius, because the ones who are crazy enough to think that they can change the world, are the ones who do."

—Steve Jobs

Personal Information Management (PIM) is the management of all the information that is relevant for our daily life. For most people this includes email messages, contact information, calendar appointments, to-do items, and all kinds of documents. The efficient management of this information is important because being able to access, change, and store information is a vital part of many work-related or personal activities.

PIM is the daily management of information

PIM, as of today, is an unsolved problem. There is much need to improve organizational habits by providing better systems for the user. Many people struggle with their daily information load and frequently misplace or forget important items, which is frustrating and can have disastrous consequences. Imagine a physician, for instance, applying the wrong treatment to a patient because the lab results were mixed up with another patient.

Four people were involved in this project

In this thesis, I will elaborate on the design process of a new system for PIM, we call the Associative PDA. I have collaborated with David Holman, Thorsten Karrer, and Jan Borchers to come up with an initial concept and to plan the progression of the project. In intermediate sessions, we have met again to refine the design according to user feedback and plan the next step in more detail. Further, all of the above have always been available for questions and guidance about the project, which I took advantage of on several occasions. Other than that, I have worked on this project by myself, designing, implementing, and evaluating all of the prototypes, as well as writing this document on my own.

The system I will introduce here is a new approach to the problem of PIM in two ways:

Mobility is an important aspect of PIM

1. It concentrates on the mobile aspect of PIM. Many PIM activities actually happen while the user is in transit. While this has been addressed by some commercial solutions like the personal digital assistant (PDA), the SmartPhone, or even many regular cell phones, only very few references are found in current literature about the importance of mobility in PIM. We believe that a successful PIM system must adhere to the problem of mobility and hope to create a new awareness with this project.

An associative network is more flexible than a hierarchy

2. It challenges the long-established use of hierarchies to organize information. In a hierarchy every information item is subordinate to exactly one element with the root element at the top of the hierarchy. In PIM the hierarchical organization is often realized using a folders and files metaphor, where each item can either be a folder, holding any number of subordinate items, or a file, a representation for an actual information item. We believe this organizational method is too restrictive to come up to the users' real needs and

propose the more flexible associative network structure to organize information instead.

Another goal of the Associative PDA is to become an example of an Organic User Interface (OUI). A system based on associations is a good premise for the realization of an OUI because the underlying model is very similar to a popular model of human memory, and it can, therefore, draw on an intuitive understanding of this model by the user.

Before talking about the project, I need to explain the concept of associations, OUIs, and iterative user-centered design, which all had great impact on the design and the progression of the project.

1.1 The Associative Network

Associations are mental connections between ideas or things. They are used in a popular model of human memory: the semantic model (see Dix et al. [1997]). Here, humans remember new thoughts by creating associations between them and existing knowledge. This way, an associative network of knowledge is created and maintained in the memory. To recall knowledge, associations are triggered and the connected information is remembered.

Associations are mental connections between ideas

In the Associative PDA, information is organized in a similar way. Information is stored as individual information items, which are representations of any piece of information, like a person or a document. These items can be directly connected to other items, using undirected connections accessible from both sides.

Figure 1.1 displays a small example network, showing the relationships between a cat, Lassie, Garfield, and Leonard Nimoy. There are two ways to find information in such a network:

Information can be found in the network in two ways: traversal and search

1. Associations can be traversed until the desired item is found: In the given example, imagine you wanted to

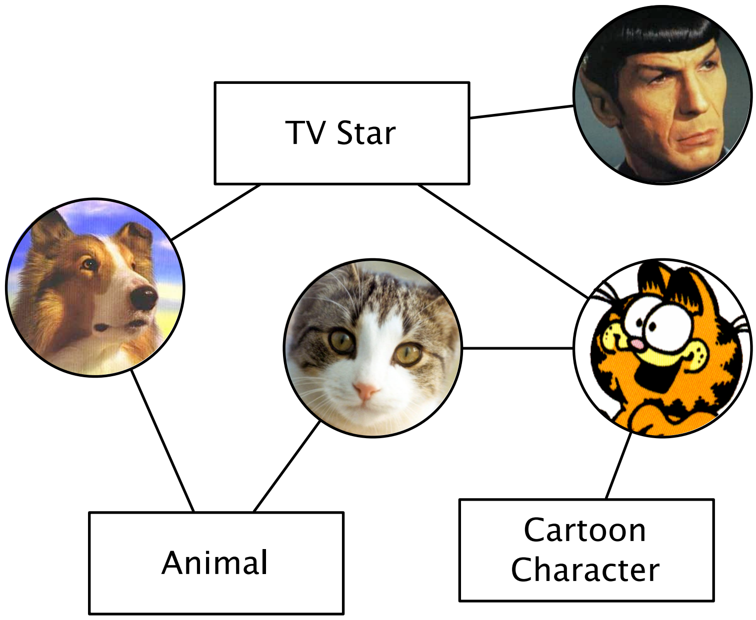

Figure 1.1: Example of an associative network

find Garfield. If you decided to start with the animals item, you would move to the cat item because you remember that Garfield is a cat. From the cat item, you could immediately move to Garfield and arrive at your destination. Instead, you could also remember that Garfield has something in common with Lassie and move over his item to the shared concept of TV Stars and finally reach the Garfield item in the next step.

2. Items can be found by identifying multiple related items and then searching only the combined neighborhoods: Now Imagine you wanted to find Lassie. You know that Lassie is both an animal and a TV star. Therefore, you would select both items and initiate a search of their shared neighborhoods. This search computes for every item in the network the cumulative distance to all selected items, returning as a result a list of items, ordered by the computed distance. I will discuss different possible measures for this distance in chapter 9. Since in our example only Lassie is directly connected to both items, it would show up

as the top result.

1.2 Organic User Interfaces

The Windows-Icons-Menus-Pointers interaction style, introduced by Douglas Engelbart in a demo of the NLS in 1968, was a major milestone for the design of graphical user interfaces and is still predominant among current operating systems (see Dix et al. [1997]). Since then, many design guidelines and rules have been proposed, for instance the Golden Rules by Shneiderman [1997], to improve the usability of user interfaces. The notion of OUIs tries to capture the essence of these principles in a new design metaphor.

OUIs follow natural laws

OUIs respect and are inspired by the natural laws of physics, biology, and human cognition. They must follow the principles of fluidity, intuitiveness, robustness, and calmness.

1.2.1 Fluidity

Follow simple rules

An interface is fluid if it follows a simple set of rules the user can always rely on. In the physical world this principle is enforced by the omnipresent rules of physics. Paper, for instance, always behaves in the same expected way. It can be written on without the text suddenly disappearing and stored without unexpectedly moving or vanishing. All physical phenomena, independent of their complexity, can be somehow explained by the laws of physics. Similarly, biological systems follow their own simple rules. Bird flight paths, for instance, appear complex yet self-organizing. Their order is always restored by simple rules that are in effect. OUIs support this notion by providing clear rules that enforce consistent constraints throughout the system.

1.2.2 Intuitiveness

Convey natural understanding

OUIs appear familiar by making use of clear affordances, natural mappings, and constraints (see Norman [1988]). Analogies are found in biological systems: Some highly poisonous animals and plants appear in bright colors to warn their predators of the poison. Over time, bright colors have come to afford poison and danger. Eventually, some animals and plants have abused this evolutionary affordance by mimicking the poisonous appearance without actually being poisonous. Conventions like these guide living beings through their daily struggle and can similarly guide human beings. To design for the human, we must understand the abilities and limitations of human cognition and find the most natural interactions and visualizations to support their activities. OUIs do so by conveying a natural understanding of the underlying functionality.

1.2.3 Robustness

Recover from errors

OUIs must be robust in that they do not break down easily and can recover from errors, possibly with degraded functionality until repair is available. In nature, if a biological system simply collapsed after a small mistake, it would die out. Instead, biological systems evolve and learn to overcome or avoid life-threatening hazards. While it might not be feasible to create software that evolves and learns to overcome its own errors, OUIs must at least be stable enough to survive any predictable situation they might be confronted with.

1.2.4 Calmness

Do not intervene

A calm interface will never interfere with the user's natural flow of work. Information output is represented in a non-intrusive way, which is immediately available if needed but otherwise not distracting. In nature, a forest conveys a great deal of information in a very calm and soothing way. The visitor can concentrate on the information provided or simply ignore it and focus on something else. Similarly, OUIs must allow the user to decide how much attention to focus on the interface.

Besides having a clear design metaphor and following the established guidelines of the field, it is important to involve the user in the design process. The next section will describe the design process I have adopted and its implication on the design.

1.3 Iterative User-Centered Design

Iterative user-centered design increases a system's usability

Nielsen [1993] describes the iterative user-interface design process and its benefits for increasing the usability of a given system. The perfect design is dependent on so many subtle factors that it cannot be found in the first attempt, if ever. Instead, designers must concentrate on evaluating and improving their design until they reach an outcome, for which the effort of further optimizing it exceeds the benefit of increased usability.

Iterative development is characterized by the DIA cycle

Iterative development is characterized by the design-implement-analyze (DIA) cycle. In the design phase, findings of the analysis are interpreted and conceptualized into the design. Afterwards, in the implementation phase, the design is implemented in the form of a prototype or final system. Finally, in the analysis phase, the system is evaluated in small user tests of usually around ten participants. The analysis concentrates on usability issues, aiming at discovering a qualitative overview rather than quantitative measure. With each iteration of the DIA cycle, the system becomes more refined and usable. Figure 1.2 illustrates the DIA cycle.

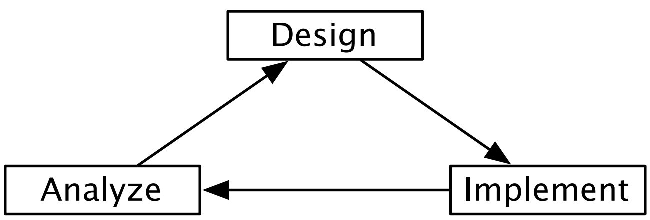

Figure 1.2: The DIA cycle

Some aspects of usability are more important than others

A system's usability consists of several aspects: ease of learning, efficiency of use, ease of remembering, error-freeness or error-forgivingness, and pleasure of use. These attributes are not necessarily of equal importance, and the focus of the development should be adjusted accordingly. Sometimes, it is sufficient if certain attribute performances are just above a bare minimum and development effort is better spent improving more important attributes. User efficiency is often seen as the most substantial attribute because of its direct impact on task performance.

For the main part of my work, I have adhered strictly to the iterative user-centered design process. I have designed, implemented, and evaluated three prototypes, of which most of the results had a strong influence on the succeeding design or were postponed for future work. In addition, I have started out with an extra analysis phase where I have conducted a contextual inquiry to learn about current behavior. I have concluded my work with another analysis phase to evaluate the final prototype in more detail and compare it to a commercial system.

1.4 Chapter Overview

Chapter 1: In the first chapter, I define PIM and give motivation why it is an important and unsolved problem. Further, I explain our approach to solve the problem and how it differs from what has been done before. Then, I describe the associative network, the notion of Organic User Interfaces, and the iterative user-centered design process, which are all important concepts that had great influence on this project.

Chapter 2: The second chapter covers the theoretical background of PIM. After providing a short historical introduction to the field, I will describe a conceptual framework of PIM activities. Afterwards, I will discuss some psychological issues of PIM and argue that it is important to understand the capabilities of the human being before designing for PIM. Then, I will present the importance of a user-subjective view on PIM and finish with explaining the problems of having multiple organizational hierarchies in the workspace.

Chapter 3: Chapter three presents several PIM design projects that have influenced my work. I will describe the Semex system, which also uses associations to store and organize information. Then, I will describe Haystack, a semantic approach to PIM, and finally UMEA, a project-oriented PIM system. All of these systems have in common, just like the Associative PDA, that they integrate all PIM activities into one application.

Chapter 4: The fourth chapter describes the contextual inquiry I have conducted by first explaining the technique in detail and then presenting the study and its results. For each finding, I argue how it has influenced my understanding of how PIM is actually performed and, consequently, the design of the first prototype.

Chapter 5: Chapter five introduces the first prototype I have created and how it was evaluated. I have drawn multiple storyboards to illustrate the initial design ideas and gain early feedback from the users. The evaluation was administrated through a life interview with an open discussion. All results are presented, and it is analyzed how the design can be changed to meet the new insights.

Chapter 6: In chapter six, I will describe the second prototype, a paper prototype, and its evaluation. This prototype's evaluation aimed mostly at gaining feedback about whether users understood the concept of associations and whether our way of visualizing and interacting with them appears natural. The results of the paper prototype yielded many interesting suggestions for design improvements and also raised some new challenges and concerns about the interactions.

Chapter 7: The seventh chapter describes the third and final prototype of the Associative PDA, a Flash prototype. With this computerized, interactive prototype, we hoped to gain insight into whether users would understand the interactions of the system and could accomplish simple tasks at reasonable performance. The evaluation of this prototype was fully automated, which allowed us to approach a larger user population and get quantitative usability results.

Chapter 8: Chapter eight describes the final evaluation of the Flash prototype, where I have compared the Associative PDA with a commercially-available reference system. I have mirrored the information space from the prototype to the reference system and measured task completion times on both systems.

Chapter 9: The last chapter discusses the outcome of the project up to the point of the submission of this document, and what should be done in the future. Several open issues are discussed and how they can be solved in future design improvements. Then, I discuss a general problem of evaluating PIM system we had faced in this project. I finish by describing the tools and techniques that have helped the most in the development of this system.

Chapter 2

Theory

"He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast."

—Leonardo Da Vinci

Origins of PIM: Memex

The origins of PIM research can be traced back to an article by Vanevar Bush [1945]. He stated that scientific papers were being produced at a rate that exceeded the available means of organization, resulting in much of the work being inaccessible. He urged his fellow scientists to come up with ways to make information more accessible. In this context, he proposed his vision of the Memex, an electronic device for storing and finding information. The Memex was supposedly linked to a library and able to search and display its content. Further, it could automatically create and follow cross-references to other documents.

The next step: Hypertext

This vision inspired Ted Nelson (see Carmody et al. [1969]) and Douglas Engelbart (see NLS Demo from 1968[1]) to independently come up with the idea of hypertext, which is one of the basic concepts of the world wide web. These technologies have had a profound influence on how we, as individuals, acquire and store our personal information.

Understand the practices by conducting user studies

Early research that formalized personal information man-

[1]http://sloan.stanford.edu/mousesite/1968Demo.html

agement activities was expressed by Malone [1983]. He had observed that the design of many PIM systems at the time was not based on a solid understanding of current organizational habits. To overcome this insufficiency, he conducted a study to determine how people organize their physical desks. This study has become an archetype for many subsequent PIM studies, for instance by Barreau [1995] or by Boardman and Sasse [2004].

PIM advancement: theory and practice

PIM research has advanced in two main areas: by creating a theoretical framework of PIM activities through observation and interpretation of PIM behavior and by designing and evaluating new system prototypes based on the theoretical knowledge.

This chapter first presents a conceptual framework of PIM activities, followed by a discussion of psychological issues. Then, I will present the importance of a user-subjective approach to PIM and finally explain the problem of multiple hierarchies in the workspace.

2.1 Conceptual Framework of PIM Activities

PIM activities: acquisition, organization, maintenance, retrieval

Barreau [1995] describes personal information management as the acquisition, organization, maintenance, and retrieval of information for individual use in a personal work environment. Because of the individuality of PIM, users develop personalized strategies to solve their unique tasks. The following is a description of the basic PIM activities and typical challenges.

2.1.1 Acquisition

Acquisition is keeping information

Acquiring information is the process of storing a local copy of or reference to an encountered information item.

Most of the time, the user decides what to include in her PIM system, and much of the included information is actually produced or adapted by the user. Exceptions are because of technical or work-related constraints, forcing the user to include or exclude certain information.

Information acquisition is chiefly motivated by need. While acquiring information, the user is immersed in traditional work-related tasks and generally not concerned with maximizing the capabilities of the system. The user's goal is to hold on to valuable information, not to store it as purposefully as possible.

Finally, information is acquired from multiple, distributed sources like file archives or the world wide web and of a large variety of formats and topics such as documents, contact details, or appointments.

2.1.2 Organization

Organization of information is the process of naming, classifying, grouping, and placing information into context to make it available for later retrieval.

Organization is placing information into context

Attributes can be used to refine the organization of information. In PIM, such attributes can be very personal and hard to differentiate, like personal importance or relevance for a certain undertaking. According to Kwasnik [1989], the purpose of the information is often the most important attribute used for organization.

Complexity of work has influence on the organization of information: Very large and complex collections are subdivided into smaller areas with adapted rules of classification. If the system fails to support this finer granularity of organization, compensating strategies like the use of suffixes in names to represent individual topics are used. Much newly acquired information, on the other hand, is important for only brief periods of time and little or no energy is put into its organization.

2.1.3 Maintenance

Maintenance is updating the structure

Maintenance is the process of identifying and processing outdated or faulty information, as well as updating the organizational structure.

Information systems grow over time, as the users add more information to them. This growth can impair on the organization and, consequently, the retrieval of information and should be compensated by appropriate maintenance activities.

Most users are not aware of the need for maintenance activities, even though they know that these mechanisms exist and how to use them. The general philosophy is: "If it ain't broke, don't fix it". If maintenance is done, it is typically motivated by external demand or a conscious effort to improve anticipated retrieval.

2.1.4 Retrieval

Retrieval is finding information

Information retrieval is the process of identifying and finding the desired information in the system, concluded by providing means for viewing and changing the information.

Retrieval activities rely heavily on the organization of the information. Information is found by identifying its location in the organizational structure, for instance by describing the position in a classification hierarchy or by identifying its attribute values.

The way in which information is identified by the user depends on her current situation. Attributes describing the context of acquisition, like time or place, can support the process of retrieval.

Information retrieval is typically initiated by a location-based probe, a jump to the general area in the organization where the item is assumed to be, followed by browsing, the activity of scanning and traversing the neighborhood around the current location. Browsing has a lower cognitive load and is typically preferred, even if faster direct-access methods are known to the user. In addition, it provides an overview of the system's content.

Finally, the retrieved information must be visualized appropriately using adaptive methods of output.

2.2 The Psychology of Personal Information Management

Lansdale [1988] shows that many of the problems and challenges of PIM are deeply rooted in human psychology and that the process of finding the best strategy can be seen as making a trade-off between available strategies with the goal of minimizing the difficulty of the task at hand. To support this process we need to understand what humans are good at and what they find difficult.

PIM is rooted in psychology

2.2.1 Human Abilities

Organization of information is achieved by classification. In most systems, this is done by categorization using natural language keywords. Natural language, though, is ambiguous: Any keyword can be used to express several concepts and vice versa. The meaning of a keyword is highly dependent on the context of its use. If the context is not preserved, categories used for storing information might not match those intended for retrieving the same information. Conclusively, there is no single best category for storing information, and finding good categories requires the laborious task of anticipating all situations where the information will be retrieved.

Keyword categorization is hard for humans

For the retrieval of personal information we need to remember certain aspects of it. Therefore, a good PIM system should exploit the strengths of human memory while avoiding its weaknesses. There are two important observations about memory: Firstly, humans find it easier to re-

Humans do not remember details and what they remember depends on the context member the meaning of information instead of its details and secondly, humans can recall only a fraction of the information stored in memory, which is dependent on the situation of inquiry. Consequently, users should understand the information and how it is organized, and the situation of information recall should match that of its storage.

2.2.2 Implications for Design

In his conclusion Lansdale [1988] provides a framework for future research and design:

Retrieval is searching, followed by scanning

> Every attempt at retrieving information involves two distinct psychological processes: recall-directed search, followed by recognition-based scanning. [...] The observed behavior with any information storage and retrieval system can be seen as a trade-off between these two processes. [...] A future system should be as near to an optimization of the two processes as possible: recall processes should allow the users to use whatever memory they have to limit the area of the database to be searched; and then the information within this area of the database should be represented in such a way as to maximally assist the search process.

Multiple keywords lead to recall versus precision dilemma

Allowing the use of multiple keywords to classify information suffers from the same problems of ambiguity of natural language and ultimately leads to the recall versus precision dilemma (see Christie [1985]): The more keywords are used to classify information, the more likely it is that the information can be found, but the greater the amount of unwanted recall will be, and vice versa. However, it seems reasonable that matters can be improved by using synonyms and semantic keywords.

Implicit meta-information holds valuable cues for retrieval

Additionally , information about the information, like size, structure, or relationships to other information items, holds valuable cues for its retrieval. This information is often implicitly visible in paper-based documents. By making it visible in electronic systems as well, the scanning process can be improved. Attributes have different purposes, though, and should be visualized accordingly. For instance, the size of a document in the number of characters is of little use for recall, since humans find it hard to remember precise quantities.

Humans do not file but the more filing is automated, the less it is remembered

Finally, a problem of recall can be seen just as well as a problem of storage: Information is lost because it is not filed properly. Humans do not spend much effort on filing activities because they do not experience any immediate payback. According to Bower [1970], however, it is much easier for humans to remember self-assigned material than that provided for them. This leads to the following dilemma: The more the user has to do when storing information, the less likely she is to do it, and the more the process is automated, the less she will remember about it. Future research should concentrate on two areas: how the storage process can be made as easy as possible without taking away control from the user and what details will be remembered if the entire system is automated.

2.3 The User-Subjective Approach to Personal Information Management

PIM is personal

One of the aspects of PIM is that it is personal, which means that the considered information is used and managed by a single user. This led Bergman et al. [2003] to suggest a user-subjective approach to PIM, advocating that PIM systems should relate to the subjective attributes the user provides. They propose three generic principles to be considered when designing PIM systems:

Classify related information under same topic

- *Subjective Classification Principle*: Classify information that belongs to the same subjective topic under the same category. Such topics are subjective to the user because the same information might be classified differently by a different user. For instance, the Roquefort Caves might be classified under "Trip to France" or "Geology Course", depending on the user. The system should allow for these personal differences and not force a universal classification in general topics on the user.

Characterize information by importance

- *Subjective Importance Principle*: Characterize and visualize information according to its subjective importance. Importance of information is subjective because what is invaluable to one person might be useless to another. This importance should be recorded and visualized properly, making important items highly visible and accessible and removing unimportant items from immediate sight.

Retain the context

- *Subjective Context Principle*: Retain the subjective context of the information. The context in which information is used is subjective to the person who uses the information. It consists of three components: external context relates to the work environment; internal context relates to the user's thoughts; and temporal context relates to the state of and plans for the use of the information. The system should capture this context, tie it to the information, and restore it when it is accessed again.

These principles have several implications on the design of PIM systems.

Employ system-wide classification

First of all, the user should be allowed to see all information related to a topic together. This implies that all information is organized under a single system-wide classification scheme that is personalized for the user.

Further, information items should be attributed with subjective importance, which defines their general visibility and accessibility. This can be achieved by changing their visual appearance, like color or size.

Finally, information items should keep references to external context, like a web page that was used to write a document, store the internal appearance when the work was last abandoned, like cursor position, and retain the state of the information by tracing previous and allowing to plan future steps.

2.4 Multiple Hierarchies in the User Workspace

PIM strategies differ between types and evolve over time

It is also important to look at current practices when designing PIM systems. Boardman and Sasse [2004] have collected cross-tool and long-term data of PIM strategies in a desktop environment. They have found that individual strategies differ between types and evolve over time. Based on the results, they have analyzed the nature of PIM sub-activities and developed a strategy classification that reflects this behavior.

Information is acquired differently between individual tools: Emails arrive automatically while files and bookmarks are stored manually. The acquired information is also valued differently: Files have the highest and bookmarks the lowest value. The amount of files and emails is similar while there are much fewer bookmarks.

Organization is not consistent between tools

Organization of information is measured by the amount of information that is filed on a daily basis and how often filing is done (immediately, occasionally, never). In their study, the underlying organization of each tool was based on a folder hierarchy. The majority of subjects employed extensive filing strategies for files. For emails filing was done generally less extensively, and for bookmarks the majority of users filed only sporadically. There were multiple organization strategies in effect for each collection and the employed strategies were not consistent between the different tools. Likely factors for the variation in these strategies are the perceived value of the information, likelihood of retrieval, availability of meta-information (like sender or subject), style of acquisition (automated or manual), and personal factors.

Maintenance is rarely done

Maintenance is done rarely for all three collections. Instead of archiving old information, it is moved out of the way, together with all other unfiled information. Extensive maintenance is mostly done during major changes in life, like new employment.

Browsing is preferred over searching

To retrieve information, browsing is preferred over searching for all collections. Files are typically retrieved using a mix of location-based browsing of folders and sorting the items in the folders. For emails, sorting and scanning of items in the inbox is the most employed strategy. Bookmarks are found similarly scanning the most recent items. Paradoxically, most of the archived information is never retrieved, but if an item cannot be found, it is highly frustrating for the user. The main reasons for such breakdowns are item deletion, clutter, and misfiling.

Classification systems evolve

All collections grew over time, but bookmarks much slower than files or emails. Similarly, all hierarchies evolved over time with by far the most changes occurring in the file hierarchy. Two of the eight participants of the study changed their PIM strategies during the study. Both changes were encouraged by reflection through the interviews. The changes were very subtle but valuable to the users.

The results have a direct implication on the integration of PIM-tools. The integrated system must support variations in the employed organizations among users and information types, and it must allow the user to changed the organization over time.

By knowing and understanding the theory of a field, much wasted time and frustration can be avoided. As Sam Levenson puts it: "You must learn from the mistakes of others. You can't possibly live long enough to make them all yourself."

Chapter 3

Related Work

"Have you ever observed that we pay much more attention to a wise passage when it is quoted than when we read it in the original author?"

—Philip G. Hamerton

Many innovative systems have been proposed for PIM. I will introduce only a small selection here, appearing especially interesting in the context of this project because they also aim at integrating all PIM activities into a complete system.

3.1 Semex

The Semantic Explorer (Semex) by Cai et al. [2005] is a flexible platform for PIM based on associations. It is built on top of an existing organizational system to provide new ways for information retrieval. Figure 3.1 shows a sample screenshot of the system.

Semex allows browsing by association

The main goals of the project are to allow browsing by association of conventional information collections and to leverage these associations to increase the user's productivity. Associations connect information items from several independent sources, creating an integrated information space

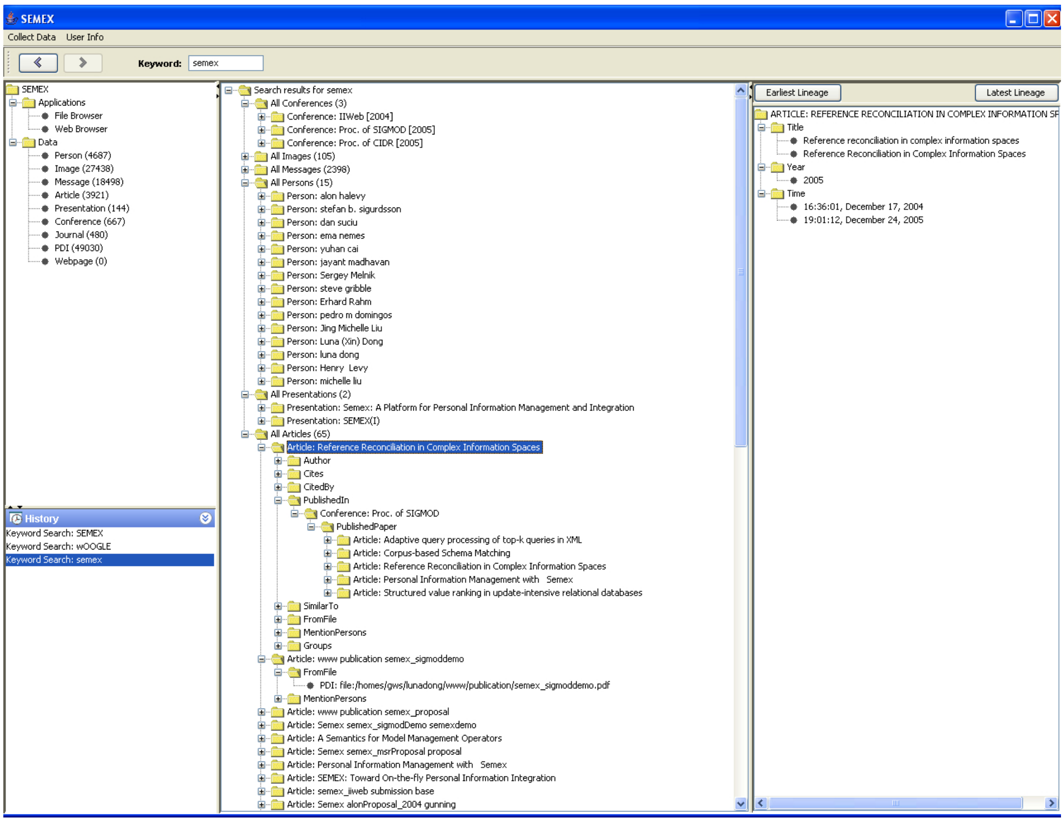

Figure 3.1: Screenshot of the Semex user interface while performing a keyword search (from washington.edu[a])

[a] http://data.cs.washington.edu/semex/semex.html

that allows the user to find and work with information independent of its source and type.

The domain model defines classes and associations

The domain model defines information classes (types) and their associations. It can be extended in two ways: by example (browsing patterns are recognized and translated into new associations) or by definition.

Associations can be inherent, extracted, acquired, or defined

Associations are created in several ways: Simple associations are already inherent in the data itself, like the contact information of a person in an address book; extracted associations are created by analyzing the content of the information, for instance by extracting author names and citations from a document source; external associations are imported from external sources like databases or web services; defined associations are manually defined by the user. The Semex system emphasizes on the automatic generation of associations in the form of simple, extracted, or external associations.

Information is retrieved in three different ways: A system-wide keyword search returns all items that somehow mention the keyword, ordered by their class; an attribute search over a specific class identifies only items of the class with a matching attribute value; and association queries allow the user to specify items by their associations. Further, the user can browse the information space by traversing associations. For each item all of its associated items are displayed and can be accessed, grouped by the association.

The Semex system is very similar to the Associative PDA: Both systems rely on associations to organize information. The major difference between the two systems is a conceptual one. The former extends an existing PIM system by proving new ways of interaction. The latter replaces an existing system with a new way of organizing information. This results in fundamentally different system requirements. For the Associative PDA it is assumed that all of the user's information is already existent in the associative network and issues like automatically generating associations and querying multiple information sources are of little importance.

3.2 Haystack

Haystack is based on RDF

Huynh et al. [2002] have developed Haystack, a platform for creating, organizing, and visualizing information using the resource description framework (RDF). It is built on top of an existing organizational system to provide new ways of interacting with the information. Figure 3.2 shows the startup screen of the Haystack system.

RDF is used to model knowledge

The RDF is a W3C specification (see w3.org[1]) of a meta model describing resources by statements of the form subject-predicate-object. It is generally used for modeling

[1]http://www.w3.org/RDF/

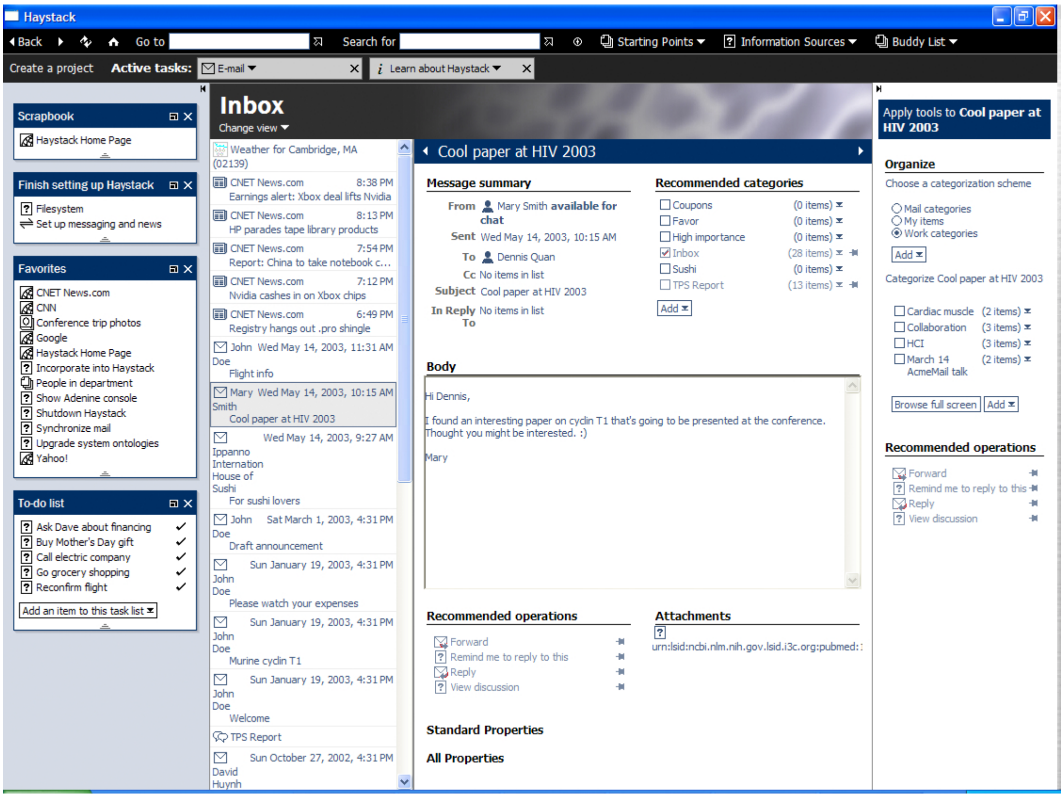

Figure 3.2: Screenshot of the Haystack startup screen (from mit.edu[a])

[a] http://haystack.lcs.mit.edu/

knowledge by expressing facts about the target domain as statements.

Haystack employs a unified classification system

The goals of Haystack are: to maximize the flexibility in how the user can organize her information; to unify all information in a single organizational scheme; to provide easy means for visualization and manipulation of information appropriate to the task at hand; and to automate certain information processing tasks using agents.

Items are organized in collections

/ The main concept of Haystack is to describe the information, its visualization, and the process of automation in a single modeling language: RDF. This model is defined in a system ontology which can be extended by the user. The ontology defines classes, their properties, how they are visualized, and the automatic procedures that can be applied to them.

Information is organized in two distinct ways. Items can be classified in collections where the interactions afford the use of multiple collections for each item. Further, direct relationships can be indicated using an integrated graph editor.

The visualization of information items is defined entirely in RDF. The user can change the visualization during runtime and adapt it to her needs. Additionally, all changes are stored in RDF, allowing these visualization solutions to be shared with other users. This also enforces a uniform user interface where all semantically identical elements act in the same way. Context menus, for instance, are defined by information type, consequently providing a uniform set of operations per interface element.

The unified structure of the information provided by RDF facilitates the creation of agents to automatically process tedious tasks for the user. To support this task, the authors have developed the scripting language Adenine, which can be compiled into RDF, allowing agents to be easily integrated into Haystack.

The semantic model underlying the Haystack system is similar to the associative network of the Associative PDA. Both models aim at representing the information space of the user in a logically concise way. The associative network is more flexible, though, by allowing direct connections between any information items without relying on a formal ontology or special collection objects.

3.3 UMEA

UMEA represents a project-centered approach to PIM

The User-Monitoring Environment for Activities (UMEA) by Kaptelinin [2003] is an application that provides the user with a project-centered approach to information management. After defining high-level activities, such as projects, the system tracks all information belonging to these activities while the user works with the system. The information can then be accessed by the high-level activity.

UMEA is informed by activity theory

The approach of this system is informed by activity theory. This has several implications for the design of such a system: The meaning of information is determined by the context of the activity and focus on context is switched constantly; activities depend on the social and physical context; and a PIM system is a mediating tool and should be transparent, preventing the user from having to focus any attention on interacting with it.

UMEA achieves these premises by allowing the user to directly indicate projects, by monitoring user activities and used resources for each project, and by organizing the monitored information automatically to make it easily accessible upon resuming a project.

UMEA monitors user activity

The application runs in two modes: The foreground mode displays projects and allows the user to view and change the organization of information by project. The background mode monitors user activities and used resources.

The author has conducted an empirical study of the system's performance, evaluating it over several weeks with eight test users. All participants but one found the system useful. One of the major problems of the system is the occurrence of multi-purpose activities, such as checking email or surfing the web, which cannot be appointed to any high-level activity. Such activities must be identified and considered separately from ordinary activities.

UMEA organizes information according to user activities. The Associative PDA supports this organization but does not enforce it throughout the system. It will be interesting to observe whether users will come up naturally with an organizational behavior that reflects this activity-centered approach.

The presented systems illustrate well the general course of designing for PIM: Researchers develop new ways of interacting with personal information by creating innovative interfaces to existing data. The Associative PDA chooses a different approach, where the underlying structure of the data and consequently the user's understanding of it is challenged.

Chapter 4

Contextual Inquiry

"Consider trying to teach someone to drive not in a car, but in a conference room. With no pedals, turn signal, or steering wheel, explain what's involved in making a turn. Try to describe what the road might look like, when to slow down, when to put on the turn signal, when to turn the wheel and how fast. It would be tempting to borrow a pie plate for a wheel and blocks for pedals. But even then, it would be so much easier to take your student out on the road and demonstrate."

—Hugh Beyer and Karen Holtzblatt

The designer is easily allured into thinking that he himself is a typical user of his system. The untruth of this is best illustrated by the following line of thought:

The designer is not a typical user

> I am a typical PIM user. I receive emails, I schedule appointments, I take notes, and I write documents like this one. I am also a typical mobile PIM user. I move around, I have multiple offices, I communicate with people while on-the-go. Why then am I not a good representative for the intended user population of my system? Because I am I. I am the designer, the person with the idea and the desire to implement that idea. Just because of that I already

> have more explicit knowledge about the to-be-designed system than any of my potential users.

This holds true especially for a field as board and individual as PIM. People adopt various strategies that vary among formats and change over time (see Boardman and Sasse [2004]). As an example, many people handle their emails substantially different from how they handle their bookmarks. This diversity of information handling strategies must be taken into account when designing a new PIM system or it will not be accepted by the users.

The contextual inquiry allows observing users in their context

Consequently, there is a clear need to investigate current work practices of PIM. To do so, we have chosen a technique called Contextual Inquiry, allowing us to interview and observe users in their work context. This, we hope, will allow us to learn what common challenges are encountered and what strategies are employed to overcome these challenges.

A guided tour reveals the structure of organization

In addition to the Contextual Inquiry, we have included a guided tour of the work space. This is a common technique for PIM studies with the aim of revealing internal structures of the organization of personal information (see Malone [1983]). Instead of concentrating on behavior and activities, the tour illustrates how information is stored and what organizational strategies are used. This information is important to determine accurate user profiles and to help understand the findings of the contextual inquiry.

4.1 Method

Beyer and Holtzblatt [1998] introduce the contextual inquiry as a field data-acquiring technique, designed to expose the structure of work, its unexpressed knowledge, and the details that have become habitual and hidden.

The contextual inquiry is based on the master/apprentice model

Interviewing an individual is more natural, when the participants can draw on a simple, familiar relationship model than on a list of rules. In the Contextual Inquiry a relationship between master craftsman and apprentice is established, where the interviewer takes on the role of the apprentice. This model is very effective as it creates an appropriate behavior for both participants: the master craftsman teaches by performing and explaining the work; the apprentice learns by observing while being immersed in the work himself. Neither participant needs to spend much cognitive effort to keep up the relationship.

The master/apprentice model is extended by four principles

The designer is not interested in performing the work of the user, though. He wants to design a system to support this work. At the same time, he cannot afford the time an apprentice would spend and often has to consider a widely varying work practice. Therefore, adaptations to the model are necessary, which are led by four principles:

Go where the user is

- *Context*: Go where the user is. Gather data inside the user's context, while preserving the workflow with only minor interruptions. Staying in context allows to experience concrete details of work and ultimately to reveal its inherent structure.

Involve the user

- *Partnership*: Involve the user. Discuss the structure of work as it gets uncovered. When something is discovered that does not quite fit, interrupt the user and ask her to explain the particular action. This reflection will eventually make the user sensitive to her own habitual behavior and allow her to actively support you.

Interpret immediately

- *Interpretation*: Interpret findings immediately. Observation discovers raw facts, which must be interpreted before they become useful. Interpretations can be incorrect though, which is why it is important to validate them with the user by sharing and discussing them.

Focus on the design

- *Focus*: Concentrate on the aspects of work relevant to the design, by establishing a clear focus and communicating it to the user. Actively direct the interview by dropping irrelevant topics without discussion. Keep in mind that focus conceals everything that is not covered by it, though. From time to time it is advisable to break out of the focus and readjust your understanding of work to match that of the user.

Running a successful interview is all about playing an adequate role during its course. The four principles and the master/apprentice model assist the interviewer in assuming such a role.

4.2 Participants

I have observed ten users in the field: a university professor, three PhD students, a shopkeeper, a lawyer, a clinic director, a staff manager, a field test coordinator, and a technical supervisor. Ages range from 28 to 60, and of the ten participants two are female. All participants are experienced computer users and have a long-established PIM system in place.

I selected the participants according to the following criteria:

Participants should be 'knowledge workers'

- The participant should be what Drucker [1973] calls a 'knowledge worker': an employee who's value to a company is based on his/her knowledge and the ability to generate new information. According to Malone [1983], people with flexible jobs such as 'knowledge workers' tend to have messier desks, which Lansdale [1988] reasons is a sign of the technology not adequately supporting the user. By concentrating on 'knowledge workers', I hope to identify ingenious and unexpected strategies adapted to work domains that require extreme flexibility.

Participants should be experts

- The participant should be an expert in his/her field with long work experience. This expertise should be reflected in the way the participant organizes his/her information. I was looking for people with a very efficient PIM system in place because I wanted to learn from their strategies.

Participants should be diverse

- The participants should be from a broad range of professions. Since PIM is omnipresent throughout any occupation, so should be my study participants.

4.3 Set-Up

Each study lasted about four hours, of which the first hour was spent explaining the method and the focus, followed by a guided tour of the office. During the tour, the participant was encouraged to talk freely about his various information collections, but I also made sure that the most common stores were included.

Participants were encouraged to talk freely

My focus was on personal information management including challenges of the current organizational tools and strategies to circumvent them.

I took only hand-written notes. Because of privacy concerns, I could not record the observation on audio or video. Much of the observed work was highly confidential and I was not even allowed to take written notes of the content of work.

During the observation, I tried to intervene as little as possible with the users. Some users tended to involve me extensively, adopting a think-aloud method while performing their work. Others were more reserved, such that I had to inquire about their current activities once in a while. Whenever I saw an activity that fit my focus, I asked the user to explain in detail what he/she was doing and why he/she was doing it this way. I wanted to understand the activity as well as its motivation in detail.

Finally, in the last 15 minutes, I summarized everything with the user to confirm its accuracy. I also invited the participant to briefly discuss some of the more interesting findings and how an improved PIM system could support the activity. It was especially interesting to hear the users' own ideas about what should be changed about their systems.

The findings were summarized at the end of a session

4.4 Findings

I have divided the findings of the contextual inquiry into three groups: (1) common activities users had to perform to work on their tasks, (2) typical strategies that were employed to achieve the individual goals, and (3) general challenges of the currently employed PIM systems.

Many findings lie outside the scope of the project

Many of the findings of the study are outside the scope of this project. I still decided to include them in this thesis because they might be interesting to fellow researchers. I will refrain from discussing any of the findings here and instead explain their influence when I am presenting the design.

4.4.1 Activities

This section describes typical activities PIM users have to accomplish. These activities can aid the construction of use cases for testing purposes and to support the design process. They can also be used as a means of motivation to demonstrate how broad and complex the field of PIM is.

Users need to plan tasks

- Users need to plan incoming tasks and remember them in time.

Users need to find information

- When working on a task, users need to identify what information is required and decide where to find the information. They need to remember whether it is already stored in their system and how it is stored, or where to look for it outside their system.

Users need to store information

- Upon encountering information, users need to decide whether the information will be needed in the future and how to store it accordingly.

Users need to work with information at varying levels

- Users need to work with their information at varying levels. In some situations, a document as a whole might be considered as one entity, while in other situations each paragraph might be relevant by itself.

Users need to share information

- Users need to share information with others. Sharing is usually limited to certain information, recipients, and time.

4.4.2 Strategies

This section lists typically encountered strategies that illustrate how people cope with their PIM problems in their current systems. Many of these strategies are likely to be compensating strategies for insufficiencies of the PIM system. They give valuable hints at how a PIM system should be extended to make it more usable.

- Tasks are planned and processed using multiple queues. These queues represent different levels of subjective importance or are motivated by technical needs. When a new task is encountered its importance is estimated and it is immediately sorted into the appropriate queue. The queues are consulted regularly to gain an overview of what still needs to be done and to decide what to work on next.

Tasks are planned in queues

- Subjective importance is dependent on personal importance, urgency, and complexity. It is approximated using meta-data like sender information in emails.

Importance is dependent on personal importance, urgency, and complexity

- Tasks that can be finished quickly are processed immediately, avoiding the queue entirely. Similarly, tasks that can be delegated are done so immediately.

Quick tasks are finished immediately

- Similar tasks are processed together.

Similar tasks are processed together

- Regular events are often remembered without explicit reminders. Therefore, they are not regarded as active information and kept out of immediate sight, even when they are approaching.

Regular tasks are remembered without aid

- Complex tasks are divided into several subtasks, which are then processed individually. If necessary, a checklist is generated as a reminder for the current state of the task and the sequence of subtasks. The state of tasks is sometimes also recorded inside the corresponding information items.

Complex tasks are divided into coherent steps

- Related information items are kept together. Distinct items, on the other hand, are kept separate.

Related items are kept together

Active information is kept accessible

- Active information is kept highly accessibly and visible to serve as a reminder of the task associated with it and to provide an overview of the information. Inactive and finished information is removed from immediate sight to avoid clutter.

Easily accessible public items are not stored

- If an information item is easily accessible from its source, it is not stored in the system but the user relies on the source instead.

The organization of information is unified

- The underlying conceptual classification scheme is unified for all information. The depth of the classification, on the other hand, is adjusted to the amount of data and might vary between the collections. Each classification system is frequently maintained: It is extended if too many items are stored under the same category and reduced if too few or no items are left in a category.

Time holds valuable cues for retrieval

- Information is ordered chronologically. Time holds valuable cues for retrieving information because users find it easy to remember when an event took place.

4.4.3 Challenges

This sections shows challenges of current PIM systems and the breakdowns I have encountered. They demonstrate the limitations of current systems and also illustrate general challenges of PIM design, which should be considered when designing a new system.

Task deadlines are not always explicit

- Tasks do not always have explicit deadlines. Sometimes the time limit is inherent in the task itself, or must be determined by the user.

Information is not always filed immediately

- Sometimes the user is not able to or does not want to file information immediately upon arrival. This information is kept in an unordered fashion, complicating the recall process, and requiring laborious maintenance activities.

- When a task is interrupted, it is hard for the user to recover her work.

It is hard to recover from interruptions

- Private and professional life is strictly separated. Many users do not feel comfortable storing private content on company hardware. Private and professional information, however, overlaps in many cases.

Private and professional life is separated

- Users must be aware of incoming tasks and messages without being distracted from their current work.

Users must be aware of events

- A task queue motivated on technical issues, like the email inbox, often separates related tasks from each other, making planning activities more difficult.

Distributing related tasks in multiple queues makes planning more difficult

Some of the findings have influenced the design

Some of the activities, like the mobile aspect, and challenges, like the unwillingness to file information, have informed the design of the first prototype (see chapter 5).

Many of the strategies, like keeping related items close and unifying the information space, have confirmed our initial design ideas. Others, like keeping active items more accessible, have inspired us to improve our ideas to come up with the design of the second prototype (see chapter 6).

Conducting this contextual inquiry has provided me with a much better understanding of the challenges and behavior prevalent in PIM. I highly recommend this technique to anyone who wants to do research in this area. There is much that can be learned from the users and their behavior, and they are eager to share it.

Chapter 5

First Prototype: Storyboards

"A lot of times, people don't know what they want until you show it to them."

—Steve Jobs

Dix et al. [1997] describe storyboards as a very simple prototypes that can be used to evaluate quickly whether the design is heading in the right direction. They are graphical representations of the physical appearance of a system without any of its functionality. Typically, they are realized in the form of a comic strip, depicting how a user interacts with the system.

Storyboards are simple prototypes

Storyboards afford high-level feedback, meaning that user will concentrate on fundamental aspects of the system and ignore small flaws of the interface. This feedback is very useful at the beginning of a design process, since the system is not entirely shaped, yet. Storyboards should be drawn by using rough design sketches to consolidate affording this kind of feedback.

Storyboards afford high-level feedback

We wanted to use storyboards to communicate our design ideas to potential users and gather early feedback about the system. This way we can ensure that we are working on real problems and that our suggested solution appears reasonable and worthwhile. Through discussion about the project, we also hope to raise early criticism of the concept and ideas for improvement. Storyboards are ideal candidates for these tasks because they can illustrate indefinite ideas and afford open discussions.

I have created six storyboards, illustrating the Associative PDA in three distinct situations, each encountered once with and once without using the device. The storyboards revolve around a young designer called Susan and her colleagues Eric and Peter. Figure 5.1 shows an example storyboard.

5.1 Situations

Situations were chosen to be realistic

The situations were chosen to appear realistic and to demonstrate the main advantages of the proposed system over conventional systems. To achieve this, they have been partly based on the findings from the contextual inquiry (see chapter 4).

Each of the following sections tells the story of two storyboards. The actual comic sketches can be found in appendix A.

5.1.1 Situation 1: Un-filed Information

Susan wants to show her colleague Eric a picture of the designs he has been working on. She remembers that Peter has sent her a photography of the current prototype via email.

Without the Associative PDA:
She opens her email client and initiates a keyword search for the project name. The search produces too many results, though, to simply go through them. She decides to reorder the list by sender and laboriously identifies the position where the Peter's emails are located. There are still many emails to go through, but she cannot think of any

Figure 5.1: Sample storyboard: "Multi-Media Madness with the Associative PDA"

other criteria to narrow down the results. She starts opening the emails one by one, looking for the picture. Eric is watching her all along, clearly getting impatient. After several misses, she gives up and describes the design of the top of her head instead.

With the Associative PDA:
She takes out the Associative PDA and accesses the project item. Since there are many items related to the project, she decides to create a combined search of the project and Peter. The resulting list is still fairly long so she thinks of other criteria to include. She is looking for a picture so it seems obvious to limit the results to pictures. This yields only very few results, which she can quickly browse and identify the correct picture.

5.1.2 Situation 2: Multi-Media-Madness

In a meeting Peter talks about several interesting talks he had attended while being at a conference last week. Susan takes notes about most of the topics because she thinks they might become relevant for her work. After the meeting, Susan is tired and heads straight home.

Three weeks later, her colleague Eric approaches her and asks her for advice on a problem. She remembers that one of the topics Peter mentioned in the meeting would probably help Eric a great deal. She cannot remember any details about it though and decides that she needs to find her meeting notes to look it up.

Without the Associative PDA:
By now, she has forgotten where she put the notes so she tells Eric that she will get back to him when she finds them.

She first searches for them in her notebook, but cannot find them. Then, she decides to search her computer. She starts by browsing her file hierarchy but does not succeed. Sometimes, she emails herself important things so she quickly checks her inbox but cannot find them there either. She also considers a system-wide search but cannot think of any keywords to use. Frustrated, she gives up on her computer and in a final attempt goes through all the stuff lying on her desk. She finally finds the notes and immediately calls Eric.

With the Associative PDA:
She takes out her Associative PDA because she always takes notes directly on the device. Since the meeting was scheduled in the device and the notes were automatically connected to the appointment, she knows that she only has to find the appointment to find the notes. She initiates a search looking for meetings involving Peter in the conference room, lying back around three weeks. The correct meeting and its notes are easily identified. She hands Eric the device to let him take a look.

5.1.3 Situation 3: Incoming Task On-The-Go

Susan has to attend a seminar out of town to learn about new design techniques. During the seminar, her colleague Peter approaches her and asks her for a copy of her meeting notes. She tells him she will send him a copy the next day.

Without the Associative PDA:
She writes down a brief reminder on her seminar notes. The rest of the seminar passes without any further coincidences. It is very late when she finally arrives home and heads straight to bed. The next day, she is swarmed with other tasks to take care of and completely forgets about the task.

A week later, she finally decides to go through her notes to prepare a presentation for later that week. When she stumbles across the reminder to send the copy, she sighs and immediately calls Peter to apologize for having forgotten.

With the Associative PDA:
She adds a new task to her Associative PDA with a short message for herself and connects it to the next day. The rest of the seminar passes without any further coincidences. It is very late when she finally arrives home and heads straight to bed.

The next day, as usual she goes through her to-do list on her Associative PDA and notices the new task to send Eric a copy of the seminar notes. Glad that the system reminded her, she gets right to it.

5.2 Evaluation

The storyboards were evaluated online while chatting with test users or talking to them on the phone. A website was created to assist the evaluation.

5.2.1 Participants

The storyboards were evaluated with eleven users

The storyboards were evaluated with eleven users: five students, two software developers, two teachers, a counselor, and a retired person. Ages ranged from 23 to 61 and of the eleven participants, four are female.

Participants were chosen to represent a broad user population of different backgrounds and experience with computers.

5.2.2 Set-Up

A website was used to assist the evaluation

The website served to organize and introduce the storyboards. Each situation is described by a short introductory paragraph and two links to the storyboards concerning the situation are provided. The storyboards were presented as high-resolution pixel-graphics, directly viewable in the web browser.

The evaluation was guided

The test users were guided through the entire evaluation. I started out by collecting user profile information and explaining the project background. Then, for each situation I asked the participants to read the introduction and then look at the storyboards. Afterwards, I asked the following questions depending on the type of the storyboard:

For storyboards without the Associative PDA:

1. Does the scenario seem plausible?
2. Has the situation described in this scenario happened to you before?

For storyboards with the Associative PDA:

1. Does the scenario seem plausible?
2. Can you imagine yourself adapting to the depicted behavior?

I encouraged the users throughout the evaluation to comment freely on the storyboards and the proposed system.

Users were encouraged to comment freely

5.3 Results

All of our participants thought that the depicted situations were plausible. Most of them had experienced the problems and liked the proposed solutions. The actual results are shown in figure 5.2.

The participants considered the situations plausible

Of the three people who had never experienced the first situation (un-filed information), one keeps his/her emails very organized, always filing incoming messages immediately and thus never has anything un-filed in his email inbox. Another person always stores and organizes email attachments in her file system, avoiding the situation just as well.

The solution that was most criticized was that for situation 2 (multi-media madness). Many participants noted that the success of this solution depends highly on the way information is entered into the system. Very few could imagine taking notes on a small electronic device, as they are available today. We avoid this concern by assuming that sufficient input techniques will be available in the future. Already, there are promising new technologies emerging like the Virtual Keyboard[1] or the Anoto Digital Pen[2] .

Participants cannot imagine entering notes on a mobile device

[1] http://www.millennia-3.com/virtualdevices/keypc.htm

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

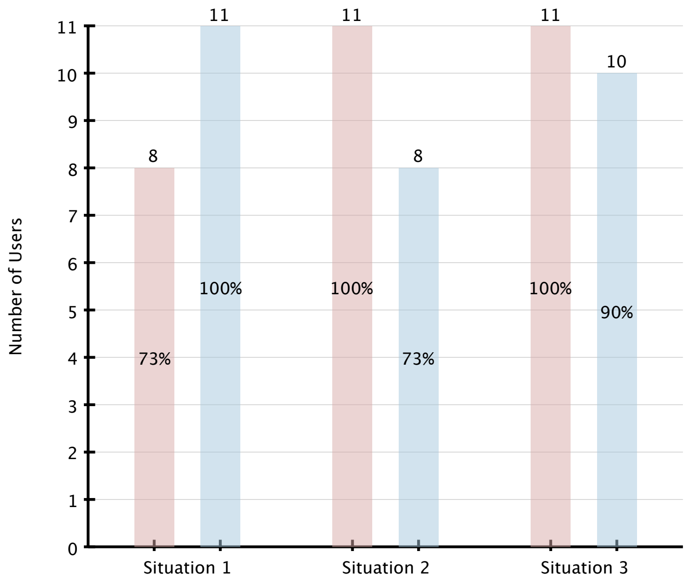

Figure 5.2: Number of users who have or could imagine the depicted situations without the Associative PDA in red and with it in blue

There are mobile PIM devices available

The one user who could not imagine himself adapting to the behavior of situation 3 (incoming task on-the-go) likes to keep all of his/her appointments in his/her head and does not use any reminders at all. Several other participants noted that they would find it hard to remember taking the device with them at all times. This problem could be solved by integrating the functionality of the Associative PDA into a common cell phone because cell phones have become ubiquitous and are rarely forgotten. Some participants, on the other hand, stated that they already acted out the depicted behavior with devices available on the market, like cell phones and PDAs, but they are unhappy with the clumsiness of interactions.

The evaluation of the storyboards has confirmed that our target problems are real problems and that our solutions appear realistic and useful. This is an important premise for any design project and should always be validated before delving into the design activities.

Chapter 6

Second Prototype: Paper

"Design can be art. Design can be aesthetics.
Design is so simple, that's why it is so complicated."

—Paul Rand

A paper prototype is an excellent means of evaluating your design at an early stage with low cost of implementation. Snyder [2003] defines paper prototyping as follows:

In a paper prototype a person plays the computer

> Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person "playing computer," who doesn't explain how the interface is intended to work.

Paper prototyping is simple and efficient

Usability testing using a paper prototype is a simple yet efficient procedure. First, the audience of your system and typical tasks are identified. Afterwards, the prototype is created by sketching the interface on a piece of paper. Finally, it is evaluated with the users by interacting directly with it. The designer plays the role of the computer, simulating the reactions of the system to user input by manipulating the paper prototype by hand.

Paper prototypes afford high-level feedback

Paper prototypes afford fundamental, high-level feedback about the design. General aspects of your interface that work especially well as well as those that have problems are quickly discovered. The unpolished look of hand-written or roughly designed sketches on simple paper incites the user to concentrate on the general functionality of the design instead of its details.

We wanted to use the paper prototype to test our initial design ideas and the basic interactions of the Associative PDA. We needed to know whether the users understand the concept of associations and how to interact with them before implementing a working prototype.

6.1 Design

The main area visualizes the current position in the network

The main area of the device visualizes the network surrounding the current position. All items related to the current selection are arranged in a circle around the center. Each item's size is determined by its importance. If there are more items than can fit on one page, they can be browsed page-wise by touching the arrows at the top.

The center is used as a temporary store and to construct searches

The center can be used as a temporary storage space. By dragging items into the center, a new representation of that item is created, keeping it accessible for future reference. The center is also used to construct a search of shared neighborhoods. By touching the empty space in the center, the search is initiated and everything related to all items currently held in the center is displayed.

The history shows previously visited items

The history bar in the middle always shows the last accessed items. It works different from the common history function in that a step backwards in the history is considered as a step forward and thus added to it. This has the advantage that the user cannot loose steps in the history by going backwards and then doing something else.

The bottom area lists attributes

The bottom area displays the attributes of the current selection, which can be edited freely.

The keyboard initiates a keyword search

Below the screen there is a keyboard, which can be used to perform a keyword search. Upon typing a keyword, the view is updated to show only related items with a matching attribute value.

Figure 6.1 shows an example screen of the prototype.

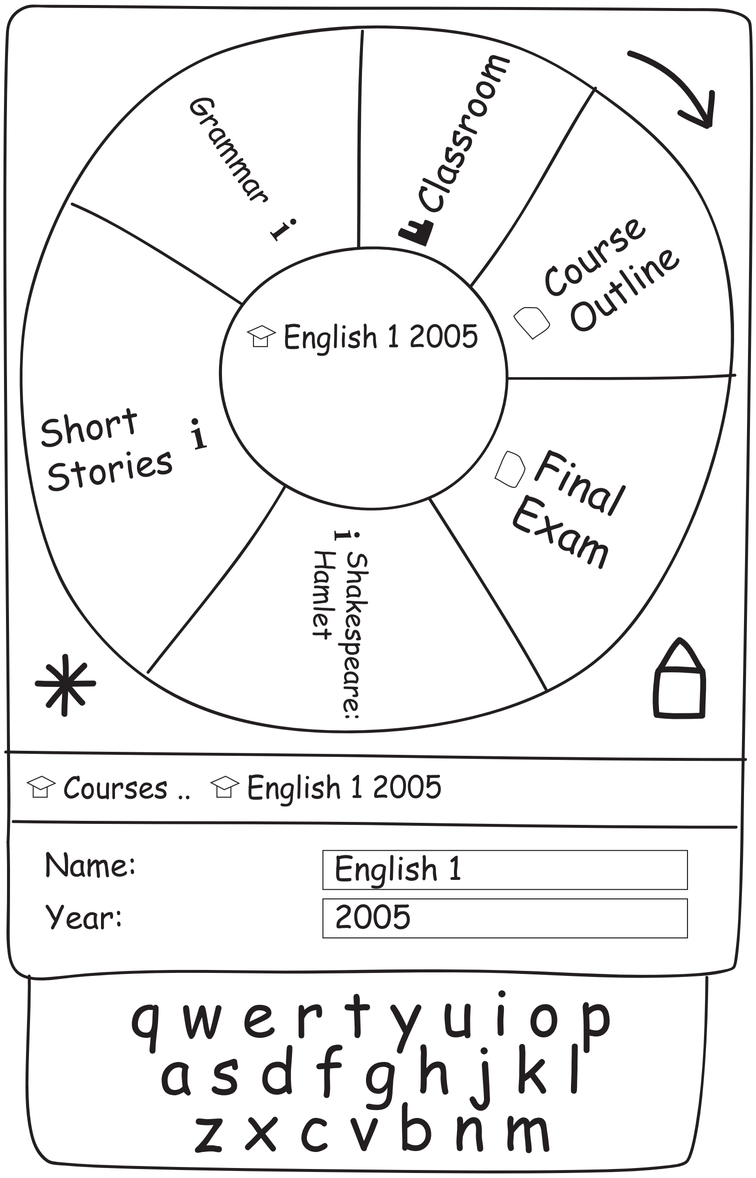

Figure 6.1: Example screen of the paper prototype

Items can be selected by simply touching them. The view is then updated to reflect the new position in the network.

Navigating and altering operations are done by tapping and drag & drop

A new association can be created by dragging an item onto another item. Associations can be removed by dragging the appropriate item out of the circle of related items. These operations, except the last, work with all representations of the items.

The star is used for copying

An item is copied by dragging it into the center while holding the star key. Similarly, items can be 'star-dragged' onto other items, creating a copy of them and connecting the copy to the target item.

The item name summarizes the attributes

An item name summarizes its attributes; it cannot be changed by the user. For instance, for items of type person the name shows the first name and the last name. This way, the user does not have to explicitly name her information items, which can be a hard task (see Lansdale [1988]).

The circular layout helps break away from thinking in hierarchies and aids spatial consistency

We decided to arrange the related items in a circle for three reasons: (1) we wanted to break away from the habits of current PIM systems, which are centered around lists and hierarchies; (2) arranging related items in a circle is a more natural visualization of a network because it simulates the visual layout of graphs and maps; and (3) we wanted to aid recall by spatial consistency, keeping items roughly in the same spatial position for the same approach.

Importance is determined by frequency of access

The importance of items is determined by frequency of access: The more often an item is accessed, the higher its importance and the larger its size. The first page always holds the most important items, providing easy access to them. The remaining pages hold the rest of the items. The ordering is always alphanumerical or chronological.

The home item lists all types

The entry point to the network is the so-called home item, listing all information types as its neighbors. When selecting a type, all items of that type are listed. This way, the user can identify items by their types. It can be accessed by touching the house icon.

6.2 Evaluation

The content of the prototype given to the users was based on an artificial data set, reflecting the information of a high school teacher. This example was chosen because it is a well-known position for many people.

The content is based on artificial data

I made sure that each participant understood the content and the organization of the data before the evaluation started. Further, the participants were assisted in any content- and organization-related questions throughout the evaluation. This does not affect the results, as we wanted to test our design under the assumption that the data set was well-known to the user.

We chose this approach because we did not see any other convenient way to test our design ideas. It is impractical to create customized, real data sets for our users since their information spaces are typically highly complex. Even though it is possible to automatically convert a hierarchical system to an associative network, the resulting data set would not benefit from the increased flexibility of the network.

During the evaluation users were encouraged but not required to describe what they were doing. We did not want to impose unnatural behavior on the user. Instead, I discussed the observations I made during the test afterwards to understand the underlying reasons and motivations.

Observations were discussed with the participants

6.2.1 Participants

I have evaluated the paper prototype with six users: two computer science students, one pedagogics student, a university professor, a lawyer, and a staff manager. Ages range from 26 to 40, and half of the participants are female.

Participants were chosen to represent a broad user base of different backgrounds and experience.

6.2.2 Set-Up

This section describes the content that was provided for the evaluations. Figure 6.2 illustrates the general layout of the data.

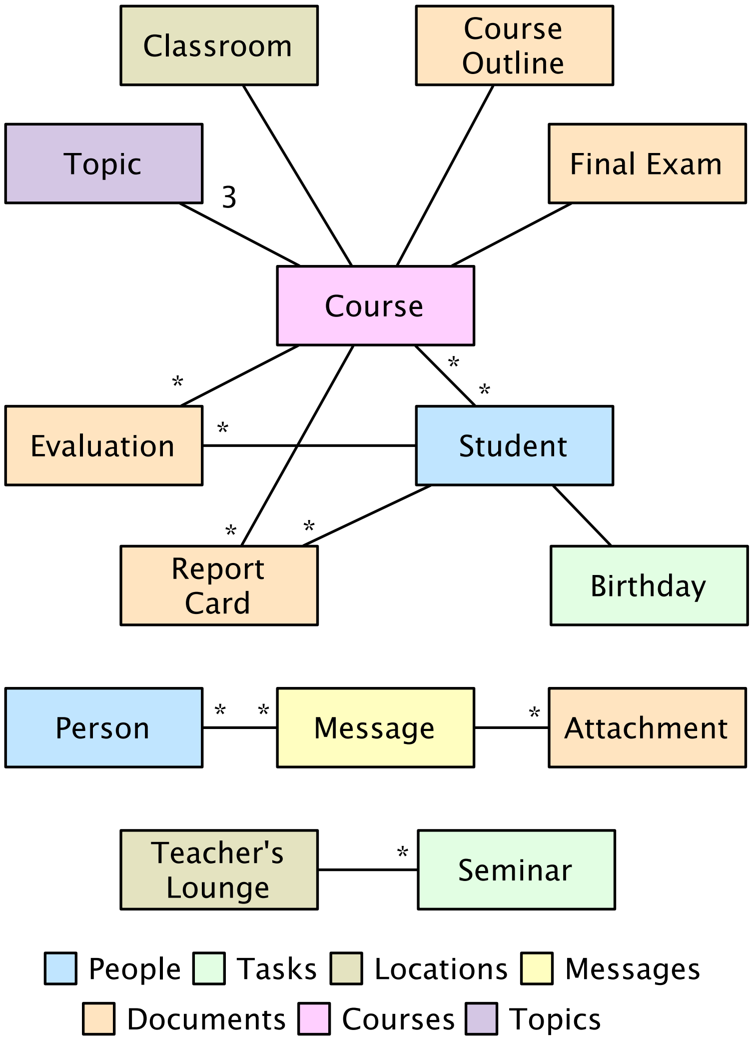

Figure 6.2: Layout of the example network. All connections, except those indicated otherwise, are 1:1 connections.

- The system is based on eight types (attributes noted in parentheses): People (name, gender), Groups (name), Tasks (name, date), Places (name), Messages (subject, text), Documents (title), Courses (title), and Topics (title).

- There are nine courses stored in the system: 'English 1', 'English 2', and 'History', each for the years 2004, 2005, and 2006. Each course is connected to two documents, three different topics, and a location. All courses, except 'English 2' in 2006 also have about 20 students associated with them. For each student and each course there are two documents called 'Evaluation' and 'Report Card' associated.
- Each of the eight topics is associated with the three courses sharing a common name: for 'English 1': 'Grammar', 'Shakespeare: Hamlet', 'Short Stories'; for 'English 2': 'Analytical Writing', 'Ellison: The Invisible Man', 'Lee: To Kill a Mocking Bird'; and for 'History': 'World War 1', 'World War 2'.
- There are six places stored in the system: 'Auditorium', 'Classroom', 'Gymnasium', 'Home', 'School', and 'Teacher's Lounge'. 'Teacher's Lounge' is connected to five task items representing seminars called: 'Seminar: Course Prep', 'Seminar: History', 'Seminar: Short Stories', 'Seminar: Teaching Methods', 'Seminar: Writing'. The dates of these tasks are set to lie at least a month apart of each other. Some of the tasks are in the future.
- For each person there is a task with a random date associated called 'Birthday'.
- There are no groups or messages, as they were not needed for the evaluation tasks.

All possible ways to accomplish the evaluation tasks were considered and the necessary views of the screen were hand-drawn onto a printed template. Figure 6.3 shows this template.

The prototype was limited to the views required for the tasks

6.2.3 Tasks

After explaining the prototype with its interactions and the

The tasks were similar to the storyboards

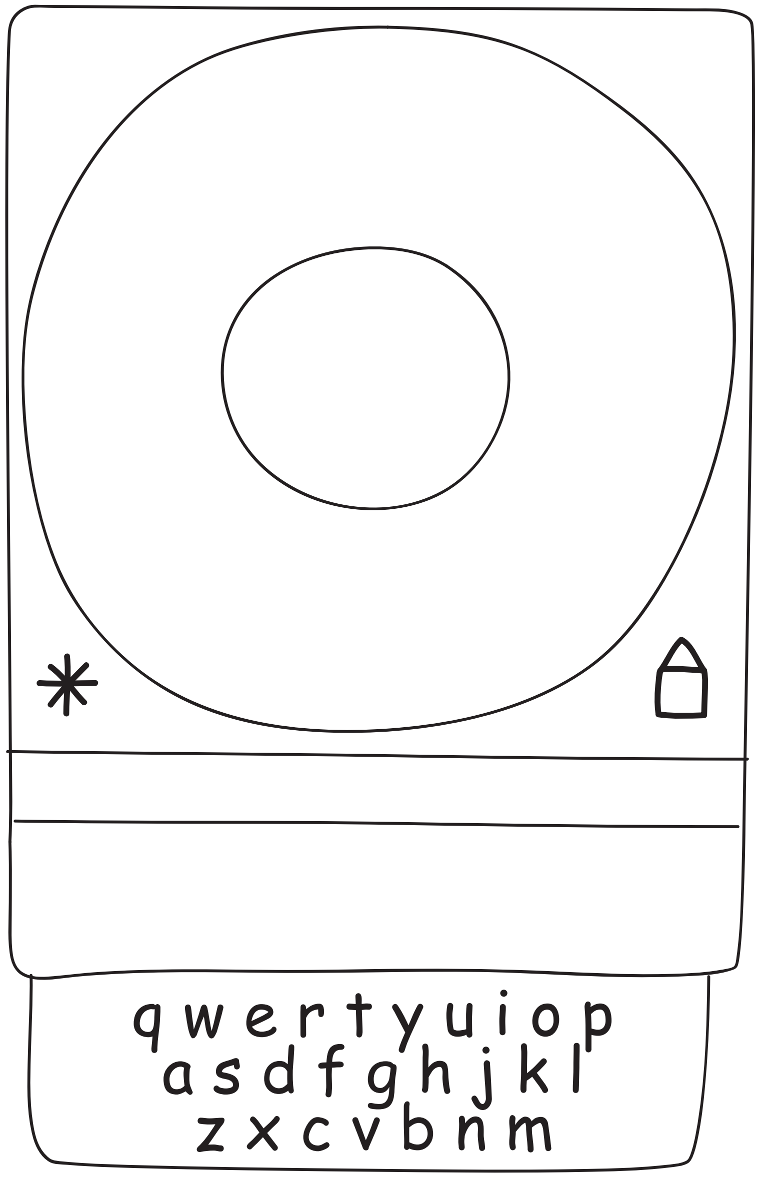

Figure 6.3: Print template for the paper prototype

data set, each participant was asked to fulfill four tasks. These tasks are similar to the stories used for the first prototype, ensuring that they are relevant tasks and supported well by the Associative PDA. The tasks are listed below:

1. *Plan Course Outline:* You want to plan your course

outline for English 2 for this year. To do so you decide to start with the course outline of the same course you held last year. You have already created the new course in your system.

Task: Please find the old course outline, create a copy of it, and connect it to the new course (English 2 2006).

2. *Prepare a new Topic:* You need to prepare the next topic in the course English 2, which will be about Analytical Writing. You remember that you have attended a seminar about this and other topics and would like to review your notes. You don't think that you have connected the seminar notes with the topic. The seminar took place in the teacher's lounge, sometime last month.

   Task: Please find the seminar notes and connect them to the topic.

3. *Parent asks for Review:* The parent of one of your students, Eric Blue, in English 1 2006 calls and wants to discuss her son's report card. This report card, along with a formal evaluation, is stored in your system.

   Task: Find the report card and the evaluation.

4. *Identify Repeaters:* The principal of the school has sent you a message that he needs a list of all students who have repeated this year's English 1 course.

   Task: Identify all students that have taken English 1 in 2005 and 2006.

The tasks are increasing in difficulty. They were chosen to require the user to understand and use all features of the system.

The tasks were chosen to cover all interactions

6.3 Results

The evaluation of the paper prototype provided only qualitative data in the form of observations and user feedback. This is typical for paper prototypes, as the interface is still fairly rough and the interactions are only simulated by the

Paper prototypes typically yield only qualitative results designer. Nevertheless, qualitative feedback is very useful at such an early stage of the design because it helps shape the design to become more usable.

The concept of associations was generally understood

Our test participants enjoyed working with the prototype. The concept of associations was generally understood and accepted. Most users did not face any problems navigating the network and finding a quick path to the desired information.

Users had problems with the search operations

The only task that caused some difficulties was the fourth task, where users were supposed to create a search over the neighborhoods of two items from the network. Three of the participants did not remember how to perform such a search. After explaining the feature again, they all mastered the task without further problems, and one participant explicitly stated how much he/she liked the elegance of this method.

The flexibility was well-liked

Several participants especially liked the flexibility of the system. This is an important factor because every person employs his/her own organization scheme. In common PIM systems, this flexibility is achieved by using workarounds like creating shortcuts to files in multiple folders.

Another user could imagine that organizing information using associations might change the way of work entirely and improve efficiency.

6.3.1 User Comments

While evaluating the prototype, many new challenges were encountered, either from direct user feedback or from observing problems the users had solving the tasks. I immediately discussed these issues with the test users to understand the greater problem. For each point discussed here, I will explain what we have later done to address it or why we believe it is not a major problem.

Time should be stored as associations

- It might make sense to store time using associations.

In the presented prototype, time was stored as an attribute. Using associations and time items instead has the advantage that the interactions for handling time are the same as for handling any other item. On the other hand, it would probably lead to an explosion of items and require special relationships between points in time and periods. I will discuss this more in chapter 9.

- The system should concentrate more on information review and authoring instead of its management because these are the most important tasks for the user. Organization of information has no immediate payback and, therefore, is not considered very important by most users. The revenue of good organization becomes visible only upon recall, but then it is invaluable. This is why we believe that a system concentrating on the organization of information is a very useful tool for the general practice of PIM. I will address how this system might collaborate with authoring tools in chapter 9.

Authoring is most important task and not supported well

- Users are used to thinking in hierarchies. They might feel lost in a network without a static organization. This issue has to be tested in future user studies using personal information sets. We hope that the users will evolve successful strategies for storing and finding information in an associative network, such that they will not feel overwhelmed by the increased flexibility any more.

Users are used to hierarchies

- In current information systems it is possible to edit a document and then store the changes as a copy leaving the original document unchanged. Users of such systems might be used to this method of interaction, which breaks down in the Associative PDA. While it is imaginable to extend the system with such a mode of editing, we believe our method is the better approach. Working first on the content and storing it afterwards is likely to result in accidentally overwriting the original content. In general, we want to avoid providing multiple ways to operate the system, following the principle of monotony (see Raskin [2000]).

Interaction order for editing content different than used to

- When intending to create an association, the user

Creation of associations might lead to premature conclusion errors might conclude prematurely when the target item is found, without creating the association. This problem is hard to avoid without introducing modes, which we want to avoid (see Raskin [2000]). We hope that this will not be a major problem and leave it open for future testing.

Search results hold valuable information

- Search results might hold valuable information, which should be stored or shared with others. Since search queries are treated just like other items, it is possible to allow storing them in the network as information items. In general, it might be useful to be able to store or share a certain subset of one's data. I will discuss some issues concerning information sharing in chapter 9.

The system has a high organizational overhead

- The system has a higher organizational overhead than common systems. This is one of the major criticisms of using an associative network instead of a hierarchy. The power of the network evolves with the number of associations created, which means that more work has to be put into the organization. We hope that the users will recognize a high-enough payoff in the improved recall, such that the system will become valuable for them. This has to be tested in future studies with personal information sets.

Separation of information not supported well

- A network structure does not easily allow separation of information into disjoint sets. This violates the finding that users tend to keep information separate from unrelated items. While it is technically possible to achieve disjoint sets, it requires extra effort and discipline by the user. Again, we hope that users will create successful strategies to overcome this challenge. To test this, a users study on personal information sets has to be conducted.

The interactions for copying items and creating associations are too similar

- The interactions for copying and linking of information items are very similar, both in their operation and visualization, and might be confused. Also, it is not very intuitive how to perform both operations on the current selection. The interaction for copying an item has been reviewed and made easier in the third prototype of the Associative PDA (see chapter 7). Instead of dragging items while holding the star button, items are copied by first selecting them and then touching the star button. The selected item is then copied and its copy is selected. Further, the selected item is now visualized in a special area in the center, making it more accessible for any operations. We believe that these changes will solve the problem.

- Users can often identify multiple ways to reach the information, which forces an unnecessary and hard decision on the user. We believe this might have been an artifact of the user study because participants were asked to explain what they are doing and tried to find the best way to solve the task instead of jumping at the first option they could think of. When working with the system for real, we hope that users will be less considerate about the way they reach their destination.

Multiple ways to achieve the goal require making a decision

- Items can have the exact same name and appearance. Sometimes this is confusing, especially when items with the same name are related to the same item or when they are both placed in the center. This issue has been partly fixed in the third prototype by displaying the most important neighbor along with the item's name. This way it is very unlikely that information items look exactly the same. Because of the limited space, this is not a viable solution for the representation in the center, which remains an open problem.

Some items cannot be distinguished

- The concept of searching the common neighborhood of multiple information items is hard to understand. This was expected as the interaction method is new and unfamiliar. We have shown in the evaluation of the third prototype (see chapter 7) that this matter can be improved with training.

Searching the network is hard to understand

- Information types can be mistaken for information items. While this is true, it does not lead to any grave errors. The system will simply refuse to create a connection between a type and an item, letting the user know of the mistake. The issue could be avoided by changing the visual appearance of the type items.

Types can be mistaken for items

- Angular text is hard to read. In the third prototype of the Associative PDA (see chapter 7), the display of

Angular text is hard to read

related items has been changed to circles with straight text inside them, avoiding this problem.

Simple operations appear cumbersome

- Simple operations like creating an association seem overly cumbersome. You first have to navigate to the first item, then drag it to the center, then navigate to the second item, and finally drag it onto the first item in the center. The main problem here is the limited space of the device, not the associative network. Similar operations on a common PDA operating system, for instance Microsoft Windows Mobile, are just as cumbersome if not worse. If users could open multiple views of the network at the same time, the operations could be realized more efficiently. This issue is apt for future research, especially in mobile input techniques and data visualization on small screens.

Mass-operations are poorly supported

- Mass-operations are poorly supported. This is a very important issue, especially for creating and managing a personal data set, and must be investigated further in the future. I will talk about how mass-operations can be supported in the associative network in chapter 9.

Privacy concerns might limit applicability

- Some workers might not be able to use the system for security reasons, for instance those not allowed to take any information outside the office. This is a general problem of mobile devices. Advanced security mechanisms might help solve this problem.

6.3.2 Suggestions for Improvement

Several interesting suggestions for improvement of the design evolved from the discussion of the breakdowns. These ideas were the basis for the design of the improved prototype, which I will discuss in the next chapter.

History should influence importance rating

- The history should influence the importance rating: Items just visited or those related to just visited items should be rated higher than others; items recently visited should be highlighted. This has been partially implemented in the third prototype (see chapter 7) by computing item importance based on access and the immediate history. I will talk about how this implementation can be improved further in chapter 9.

- Allow a system-wide keyword search originating from the selected item with its results sorted according to some distance measure like the sum of squared distances. We intentionally do not want to include a system-wide search in any prototype because we fear that our users would overly rely on this feature and ignore the features of the associative network. I will discuss in chapter 9 how such a search could be realized.

Integrate system-wide keyword search

- The number of related information items should be visualized when there are multiple pages. Further, the interaction style of browsing page-wise is slow and cumbersome. This has been improved slightly in the third prototype by visualizing the number of pages (see chapter 7). In chapter 9, I will discuss how this could be improved further using interaction techniques like the fisheye view.

Multiple pages are poorly visualized

- Automatically generate associations when creating new information items, for instance to recently visited items or the cloned item itself. These associations should be valued less than those added manually. This approach, just like any automatic generation of content, will help reduce the workload of the user while managing her information. With this project we want to follow another approach though, where we support the user as good as possible while leaving her in total control. Please see chapter 2 for a discussion of both approaches.

Associations should be generated automatically

- Include a full screen edit and review mode for documents. This feature is not necessary for the evaluation at this stage because it does not illustrate any of the features of the associative network. I will discuss in chapter 9 how it could be implemented.

Include a full screen authoring mode

- In large sets, classify items by common attributes and relationships. For instance, the gender of a person is always either male or female, providing a powerful classification option. For locations or times, on the other hand, the user could start with a very broad

Provide adaptive classification by common attributes and relationships information item like a school or a month and then move to more specific items. I will discuss this topic and its implications in chapter 9.

Include versioning

- Include versioning of items and associations. Whenever an item or association is changed, instead of overwriting the original, store it as a new version. This feature is already available for common PIM systems in the form of concurrent versioning systems like SubVersion[1] . I will discuss in chapter 9 how this could be integrated into the Associative PDA.

Allow the construction of a hierarchy from the network

- Allow the construction of a hierarchy by defining an arbitrary root item. This makes sense for items that are part of a logical hierarchy like locations. How this could be implemented will be discussed in chapter 9.

Port the system to the desktop computer

- Create a similar system for the desktop computer. The concepts of an associative network could be used in any computing system. We want to concentrate on a mobile device though because we believe the mobile aspect provides additional benefit for the users.

The results of the evaluation have yielded many problems of the device and promising solutions to improve the design. In general though, it is apparent that our users understand and accept the concept of associations.

Interacting with the system takes getting used to but is eventually understood by the users. Conclusively, we have decided to include a tutorial, explaining the concept of associations and the interactions of the system, in the next prototype.

As a next step, we want to evaluate the performance of our system and its interactions. To achieve this, a more advanced, working prototype is needed.

[1] http://subversion.tigris.org/

Chapter 7

Third Prototype: Flash

"I saw the angel in the marble and carved until I set him free."

—Michelangelo

Adobe Flash (formerly Macromedia Flash) is a multimedia authoring tool for web applications. It allows the designer to create interactive vector animations by sketching individual pictures in frames and creating transitions between them. A scripting language called ActionScript is used to implement the interactive aspects of the animation.

Flash is a common prototyping tool

Preece et al. [2002] mention Macromedia Director, a more powerful variant of Macromedia Flash, as a common high-fidelity prototyping tool. Flash prototypes can be very elaborate representations of the design. It is possible to create convincing interfaces, for instance by using screenshots of actual user interface components, and to make them responsive to user input. Typically, flash prototypes concentrate on the user interface and ignore any underlying functionality. They demonstrate how the user would experience using the system without having to implement the whole system.

Flash prototypes afford low-level feedback

The goal of the Flash prototype is to evaluate the interface and its interactions in detail. Because of its convincing appearance, it appears very similar to a finished system and affords low-level feedback about details of the design. The user assumes that the system's concept is finished and, therefore, does not question it. Instead she will concentrate on specific interactions and give feedback about the feel of the application.

Our goal was to create a convincing demonstration of the Associative PDA using a Flash prototype to evaluate the details of our design and, at a later point, to compare it to competitive systems. This chapter describes the design of the prototype and its evaluation. The comparative study is presented in chapter 8.

7.1 Design

The design of the Flash prototype is based on the paper prototype

The appearance of the Flash prototype is very similar to that of the paper prototype. The general layout has stayed the same. The graphics were overhauled to come across more polished and the background has been changed to a gray color to highlight the items. Figure 7.1 shows a screenshot of the Flash prototype.

The related items are now displayed without angular text

The items in the main area are now displayed in circles, arranged around the center. The size of the circles is changed according to the item's importance. Its type is illustrated in color, making different types stand out more. Below the name of the item the most important neighbor's name is shown. This feature has been motivated by user feedback from the evaluation of the paper prototype. The buttons to browse multiple pages now include an indicator, visualizing the number of pages and the current position. This is achieved by displaying a line for each page next to the appropriate button. Browsing of pages is visualized by rotating all related items in the respective direction.

The center now includes the current selection

In the top of the center area the current selection is shown and can be interacted with just like any other item representation. The area below can be used as before to temporarily store items for future reference or to construct a search of combined neighborhoods. On the right side a button in the shape of a magnifying glass has been added, which will

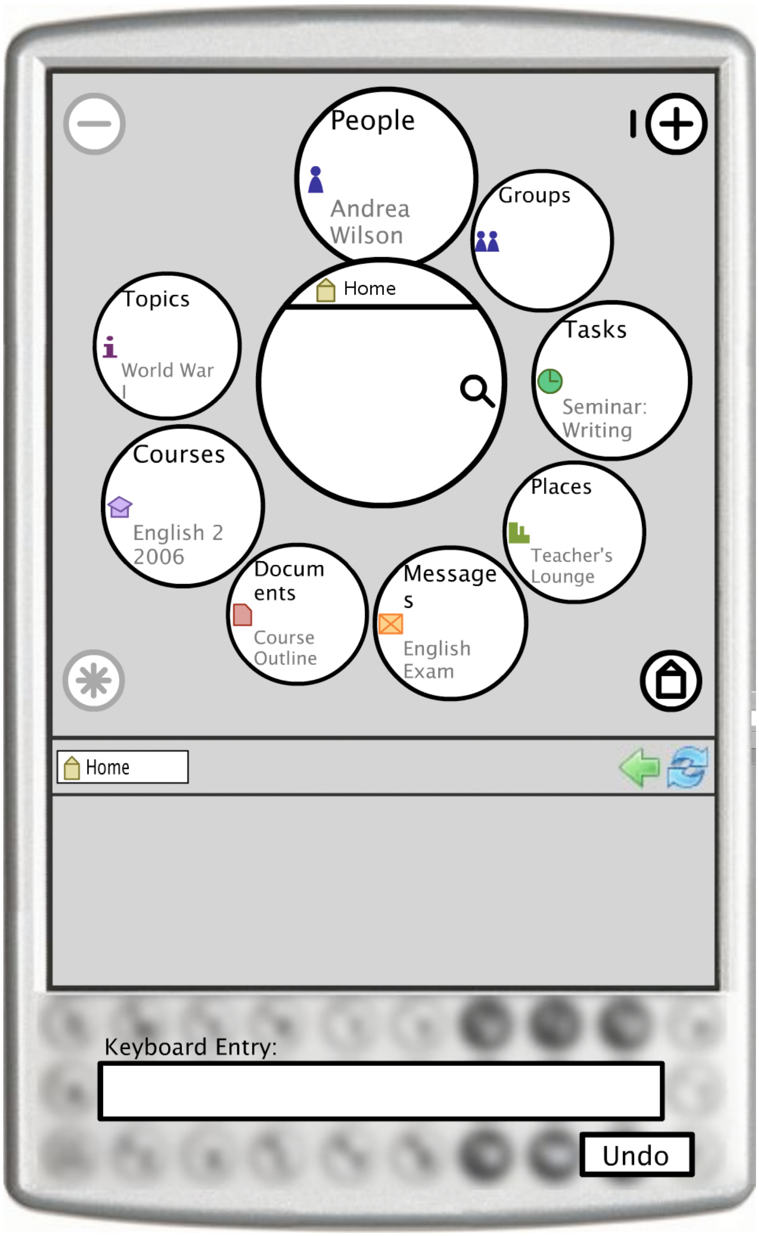

Figure 7.1: Screenshot of the Flash prototype.

trigger the search if clicked.

The history bar in the middle does not show the current selection any more. Adding a new item to the history is visualized by moving all existing items one slot to the left and

The history does not display the current selection any more then placing the new item on the rightmost position. Two buttons have been added on the far right: a back button to move one step back in the history and a refresh button to refresh the current view. The first operation was included because of user feedback; the second because of an implementation problem where sometimes the interface is not refreshed properly. This problem is explicitly mentioned in all task descriptions for the evaluation.

The details, keyword search, and basic operations are unchanged

The details area at the bottom has remained the same as in the paper prototype. The value fields have been realized by editable text fields. Changes are applied immediately without pressing a confirmation button.

The keyboard entry to perform a keyword search has been realized using a text field. Upon typing the keyword the main screen is updated immediately to show only related items with a matching attribute value.

Items are selected by clicking them. They can be placed in the center by dragging and dropping them there. All dragging operations are visualized by displaying the name of the item next to the mouse cursor while dragging. Associations are created by dragging an item representation onto another. They are removed by dragging the item from the main area anywhere onto the gray area. If a type is selected while removing a related item, the item is deleted entirely.

An undo function was included

Since fatal mistakes, like accidentally dropping an item on the gray area instead of the target item, are likely to happen with this set-up, we have included an undo function to recover from such errors. It is triggered by clicking a button at the bottom of the device. Only altering operations, like removing or creating associations or items and changing attributes can be undone.

The star item now copies the selected item

An item is copied by selecting it and then clicking the star button. This change was motivated by user feedback from the paper prototype evaluation.

Importance is now dependent on frequency of access and last visit

The importance of an item is computed by formula 7.1. It is determined by the number of times an item is accessed and the time it was last accessed. By giving the former more weight, recently visited items are highlighted by being displayed in a larger circle than other items.

$$importance = \max\left(0.9 \cdot \left(\frac{count}{max_{count}}\right), 1 - (age/60)\right) \quad (7.1)$$

with:

- $importance$: importance value of the item, $importance \in (0,1)$
- $count$: the number of times the item has been accessed
- max_{count}: the number of times the most accessed item of all related items has been accessed
- age: the time between now and the last time the item was accessed in seconds

7.2 Implementation

The prototype was implemented in Adobe Flash and makes use of a PHP back-end. Communication between front-end and back-end is done via XML. The network with all items, associations, attributes, and the history is stored in a MySQL database. Figure 7.2 illustrates the implementation layers and their communication.

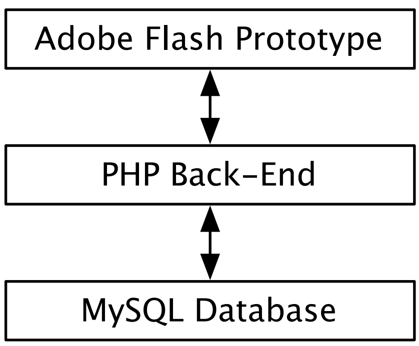

Figure 7.2: Implementation layers of the Flash prototype.

7.2.1 Adobe Flash Front-End

A central object controls most of the communication

The Adobe Flash front-end was realized using the scripting language ActionScript 2.0. A global object of the class MainController takes care of most of the communication with the PHP back-end and the creation of visual objects like the related items and the history items. The MainController provides functions to select a certain item, go back one step in the history, perform a keyword search over the selected item, create a copy of an item and select it, and undo the last change. All of these operations are accompanied by a refresh of the screen, which calls the get.php function of the PHP-backend, parses and stores the resulting XML string, and creates appropriate interface objects for the returned information.

Items are realized as specialized objects

All item representations are objects of specialized classes (CenterItem, HistoryItem, RelatedItem, SelectedItem), which manage all operations, like drag and drop manipulation and navigation. This way the interface object can decide for itself what to do depending on the operation and the target.

Animations are realized in Flash

A few animations were included to make the device appear more fluent. Whenever the user browses the related items pages, the view is rotated 360 degrees in the appropriate direction to create the illusion of moving through the item space. Further, whenever a new item is added to the history, the existing items are moved to the left with the first item disappearing from the screen to indicate a forward movement in the history. These animations were realized in Flash itself and are triggered by the corresponding ActionScript object.

7.2.2 PHP Back-End

The scripting language PHP was chosen for the back-end because it provides the necessary XML functions to efficiently communicate with Adobe Flash and the author was already familiar with it. All parameters can be given using either a GET or a POST HTTP request.

The back-end provides the required functions for the Adobe Flash application to query and alter the database. This extra layer is necessary for two reasons: (1) Adobe Flash does not come with integrated MySQL database support and (2) since Adobe Flash runs on the client machine, the database communication would have to be done over the open internet, which is a security risk and inefficient.

The back-end mediates between the database and the front-end

The back-end consists of three functions:

get.php

The get function retrieves the network information around a selected item and returns it in XML form. In addition it is used to add new items and to perform an undo step. It accepts four parameters:

get.php retrieves the network information

- i: defines the database ids for the item to be selected. If multiple items are provided, using a comma to separate the values, a search of the shared neighborhoods of all items is performed. If no item id is provided the last accessed item from the history is selected.
- s: defines the keyword for a keyword search.
- c: tells the system to copy an item. If its value is 1, a copy of the selected item (provided by parameter i) is created and returned as the new selection. For the home item ($id = 0$) or a search, an exception is triggered. For type items ($id < 10$) an empty item with the according type is created. The use of this parameter creates a new undo step.
- u: tells the system to perform an undo step. If its value is 1, the last undo step from the database is retrieved and applied. Finally, the item from the parameter i is selected and returned.

The parameters s, c, and u must not be provided at the same time. After processing the parameters, the counter of the selected item is increased and the item, its related items, and its attributes are retrieved from the database. Afterwards, the importance is computed for each related item and they are arranged appropriately. Finally, the history is retrieved from the database.

The output of the script is all the above information in XML form. Figure 7.3 illustrates the layout of the XML document. This output can be easily parsed by the Adobe Flash application.

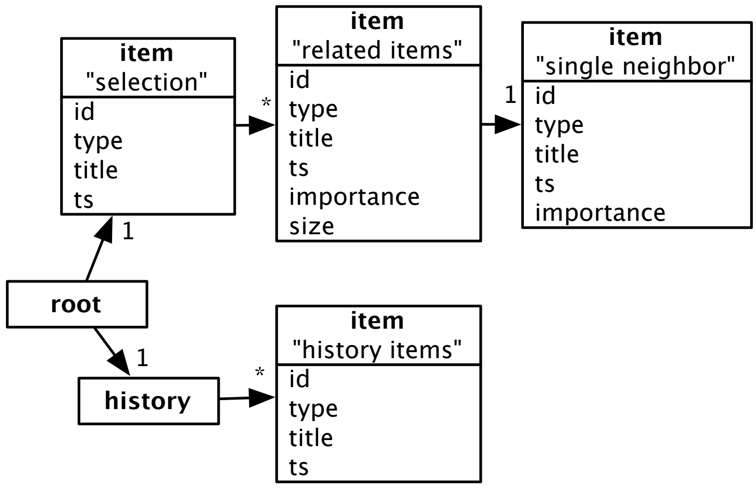

Figure 7.3: XML Layout of the return string of the get function.

assoc.php

assoc.php manipulates associations

The assoc function is used to manipulate associations. It accepts two parameters:

- *i*: must contain two existing item ids, which are separated by a comma. The script then creates an association in both directions between the items represented by the two id values.
- *d*: must contain two existing item ids, which are separated by a comma. By invoking this parameter, the script removes any associations between the items represented by the two id values. If one of these items is a type item, the other item and all of its attributes and associations are deleted.

Only one parameter can be provided at a time and there is no return value. This function creates a new undo step.

attr.php

attr.php manipulates attributes

The attr function is used to change the attributes of an item.

It accepts three parameters:

- a: defines an attribute id to be altered. If this parameter is provided, the attribute represented by this id value is changed.
- i: defines an item id for which a new attribute is to be generated. If this parameter is given, a new attribute will be created and connected to the item represented by the id value.
- k: defines the new key name for the attribute.
- v: defines the new value for the attribute.

Either the parameter a or i must be provided, but not both. Just like the previous function, there is no return value and a new undo step is created.

7.2.3 MySQL Database

The associative network is stored in a relational database on a MySQL server. For all tables the InnoDB engine was used, allowing foreign key constraint checks. This ensures data integrity by verifying the existence of foreign keys and triggering automatic update operations on all tables with reference to a key upon change. The database schema includes seven tables:

The database holds the associative network and the history

- *types*: Types holds all information types available in the network and their respective names. For this purpose, the field 'type_name' stores the name in plural as displayed on the type node itself and the field 'item_name' stores the name in singular as displayed for each individual item.

Types holds information types

- *items*: Items stores all information items in the network. The type of an item is stored as a reference to the types table. In addition, each item has a creation time stamp, a title summarizing the main attributes,

Items holds information items

and an access counter. The creation time stamp is created by the database upon creation. The title and the counter, on the other hand, must be updated manually. There are several special items: The item with $id = 0$ is the home item, items with $id < 10$ are type items. These items have an empty type and have no associations or attributes. They solely serve to store the names of the home and type items.

Associations holds associations

- *associations*: Associations connects items with each other. Each association stores references to two distinct information items, creating an association between them. For performance reasons, the stored associations are directed such that all related items can be determined using a simple select query. The backend is in charge of ensuring bidirected associations.

Dattributes holds the default attribute keys

- *dattributes*: Dattributes holds the default attributes for an information item, depending on its type. A reference to the types table is stored for each record along with the key of the attribute.

Attributes holds attributes

- *attributes*: Attributes stores the attributes for all information items in the network. Each item can have multiple attributes, which is achieved by including a reference to an item for each attribute. In addition, the key of the attribute and its value is stored. This method was chosen to achieve maximum flexibility in the choice of keys for attributes and not be restricted by a default set.

History holds the history

- *history*: History stores the items the user has previously accessed. This is done by storing a weak reference to the item. Weak means that no foreign key constraints are checked, as items might be deleted after they have been accessed, but the history record should still remain in the database. This solution is simply a quick solution to the problem and should not be adapted by the final system.

Undo holds undo queries

- *undo*: Undo holds all information necessary to undo altering operations. This is achieved by storing SQL queries to recover the changed or lost data. Multiple queries are stored in multiple entries of this table, with all records belonging to one undo step having the same value for the 'step' field.

Figure 7.4 illustrates the database layout.

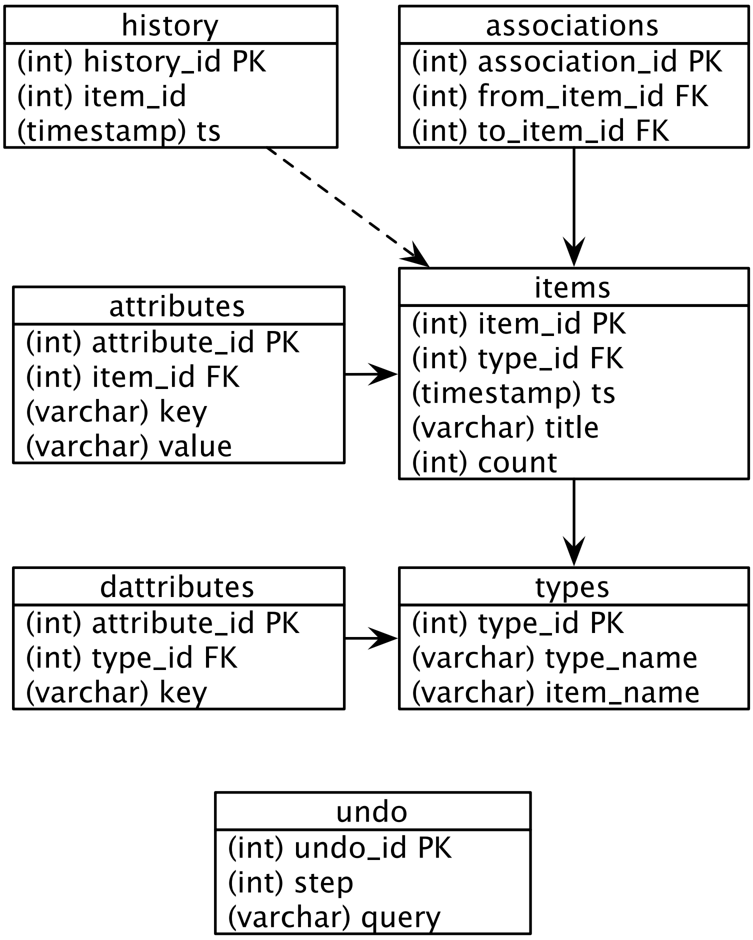

Figure 7.4: Database layout of the associative network. Solid arrows indicate relationships; dashed arrows indicate weak relationships.

7.3 Evaluation

Since an Adobe Flash prototype can be easily distributed and tested with a broad user base over the internet, we decided to create an automated online evaluation using a website.

The evaluation was done online

This decision was partly motivated by the results of a study

the software of mobile devices can be remotely evaluated by Waterson et al. [2002], where the authors have evaluated how remote usability testing performs in comparison to attended testing for mobile devices. They have found that remote testing can successfully identify web content-related usability issues, while it is not useful for identifying device-related issues. We saw this as a confirmation that this method of testing would provide us with good data, because we were not interested in the performance of the actual device, but how the interactions were understood and executed. The issues we wanted to identify were content-related and, because of the set-up, fairly similar to web-content.

The content was similar to that of the paper prototype

For the content, the example of a high school teacher's information space was chosen again (see chapter 6) and slightly modified. The report card document, connected to each student of a course, was replaced by an information item of a new type 'Grade' with one numerical attribute, representing the actual grade for the student and course. This was changed due to general user feedback, because a report card was understood as an overview of all grades of a student for a whole year. Further, a few unimportant items like the 'Final Exam' document were removed to make the network less complex and easier to explain.

The evaluation consisted of three different aspects: the success and completion time of tasks, user comments, and a short survey at the end of the evaluation.

7.3.1 Participants

Of the 22 participants, five stopped the evaluation before even reaching the actual tasks and five more did not finish all tasks, leaving only twelve successful evaluations. Only these twelve participants are considered in the results.

Half of the twelve participants are female and ages range from 23 to 37. Occupations include eight students of varying fields of study (3x computer science, medicine, technical communication, pedagogics, electrical engineering, cognitive science), a staff manager, a university professor and two application developers.

The evaluation was open to anyone who wanted to participate. It was announced via email mainly to friends and coworkers.

7.3.2 Set-Up

A website was created to guide the participant through the entire evaluation. It was tested on Microsoft Windows running Internet Explorer 7.0 and Mozilla Firefox 1.5 and Mac OS X running Safari and Mozilla Firefox 1.5. Both systems had the latest Flash player installed.

A website was used to guide the participants

The website was realized bilingual in German and English to allow international participation without impairing the local population. At the beginning of the evaluation the user can choose her language of preference.

The website was bilingual

Afterwards a short text explaining the background and purpose of the evaluation is displayed. The text also explains the concept of associations and introduces the data set.

On the next page the participant enters some general profile information. Fields include gender, age, occupation, and experience with computers. In addition the participant is asked whether she has participated in the previous evaluation of the paper prototype of the Associative PDA.

After this information is entered, the user is presented with the Flash prototype and seven tutorial tasks to guide the user through the interface. The user can navigate the prototype freely but gets very clear directives on what to do. The tutorial aims at explaining the basic and advanced interactions of the prototype and providing an understanding of how the associative network works. Whenever a tutorial task is successfully accomplished using the Flash prototype, the website is notified and advances to the next task, letting the user know of the success. At any time, the participant can leave textual comments. It is also possible but discouraged to skip the entire tutorial.

A tutorial introduced the device and its interactions to the user

After the tutorial, the user is presented the actual evaluation tasks, one at a time. Similar to the tutorial, the participant can navigate the Flash prototype freely, but is now confronted with real PIM tasks and has to figure out the individual solution by herself. Again, as soon as a task is fulfilled the website automatically advances to the next task and lets the user know about the success. Comments can be left at any time and Each task can be aborted to avoid getting stuck. For each task completion time, history, and final state of the network are stored.

At the end a short survey is presented

Finally, the participant is asked whether she enjoyed working with the prototype and if she would consider using the proposed system in her daily life. Additionally, the user can leave more textual comments about the system before finishing the evaluation.

7.4 Results

The majority of participants enjoyed the system

The results discovered many smaller flaws of the interface and some fundamental problems of the system. Still, the majority of users enjoyed working with the system and would consider using the final system for themselves (7 out of 12 for both). On a scale of -2 to 2, with -2 strongly negative, 0 undecided, and 2 strongly positive, our participants voted on average 0.58 on whether they enjoyed the system and 0.5 on whether they would consider using it.

Only 70% of all tasks were completed

Only about 70 % of all tasks were solved and task completion times vary extensively between the users. Especially tasks 1 and 4 have extremely high standard deviation values. The results of the statistical analysis are shown in figures 7.5 and 7.6.

There are several reasons for the low number of solved tasks:

Participants did not read the tasks thoroughly

- Two participants did not read the task description thoroughly and therefore performed the wrong task. For instance, both copied the 'English 2 2005' course item instead of the 'Course Outline' document item in task 2.

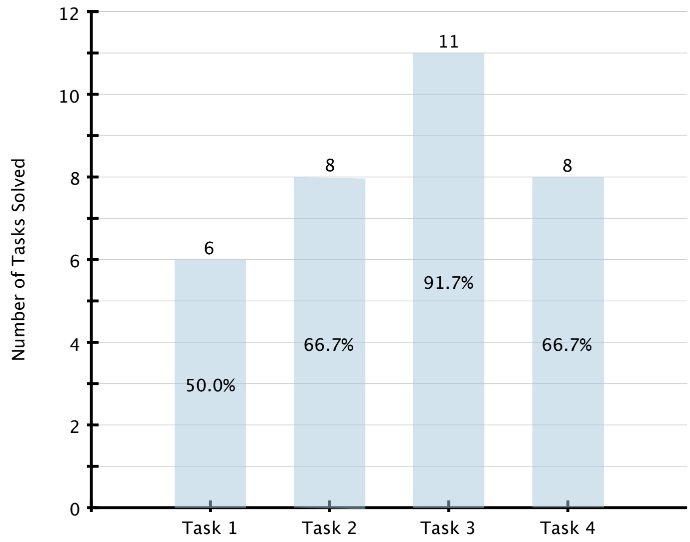

Figure 7.5: Number participants who solved a task

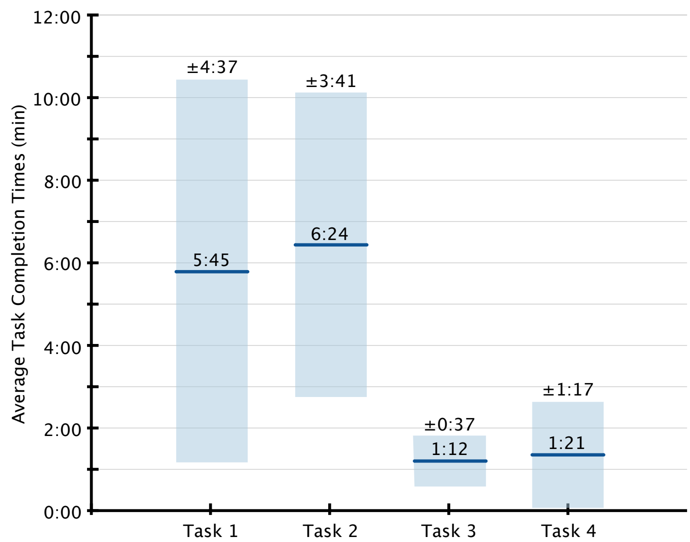

Figure 7.6: Average completion times and their standard deviations

One participant accidentally changed the network

- One participant could not find the 'Teacher's Lounge' item. It is likely that the user has accidentally deleted the item during the tutorial.

One participant finished a task in a way that was not detected

- One participant has identified the students connected to both courses 'English 1 2005' and 'English 1 2006' by hand, which the system did not accept as solving the task.

Participants had problems understand the interactions

- Two participants had problems understanding the interactions for copying an item and creating associations, which prevented them from solving several tasks. It seems that the tutorial was insufficient for teaching these interactions.

Participants felt lost

- Two participants felt lost in the network and skipped some tasks out of frustration.

Most of the reasons for failed tasks are not due to problems of the design

Only the last reason is a problem of the associative network. It appears that for some users the visualization of the network causes problems: First, the current selection is not noticeable enough and should be more clearly distinguished from temporary items. The users feel lost when they do not know exactly where they are. Second, information items and type items are indistinguishable but can only be navigated from type to item. This behavior is inconsistent and confusing. Third, the transition between items is sudden and confusing. The animation of the history bar is distracting at best.

The other problems were caused by flaws of the evaluation method, mainly insufficient training. Since we wanted to examine the system under the assumption that the interactions are known to the user, they can be ignored.

The high variance of the results is probably caused by the unattended evaluation style

The high variation of task completion times indicates that the results are not very reliable. We blame this outcome mainly on the unattended evaluation style. For once, users were struggling with the interactions of the system, wasting time on what should be known. Further, users probably took breaks during the evaluation, which could not be tracked by the system. On several occasions the participant remains at one information item for several minutes, which could be interpreted as a very long consideration process or, more likely, a break.

Some participants performed very well

Despite this, it was very reassuring to see that some users performed very well on some tasks. Figure 7.7 shows the best times for all tasks. As a reference, the author has measured his own performance to simulate that of an expert. Considering that the users had to read and understand the tasks unlike the author, a difference as low as 15 seconds indicates that this user performed about as good as the author.

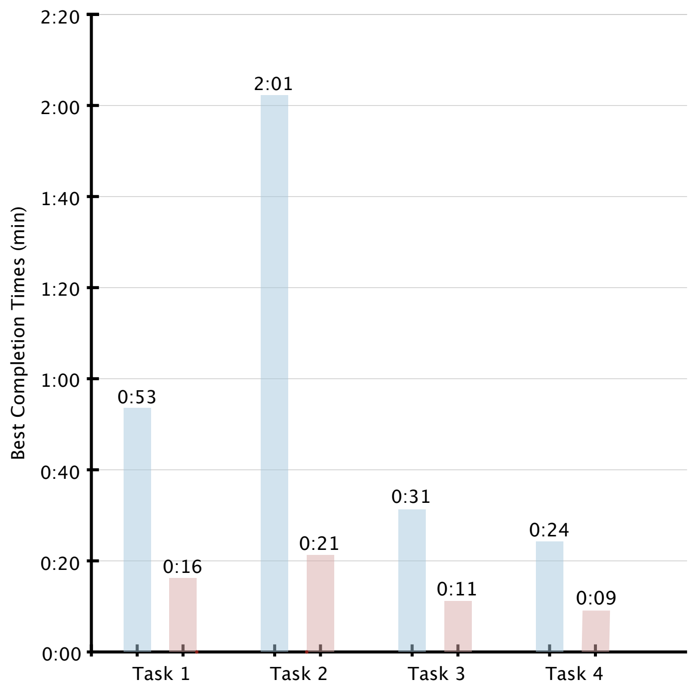

Figure 7.7: Best task completion times of the users in blue and the author in red

7.4.1 User Comments

The most useful results yielded the user comments during and after the evaluation. This section lists all comments and a quick discussion of how the described problem can be solved:

The automatic update of text field values is unfamiliar

- The of automatic update of attributes is unfamiliar. The participants expect to press a button to update and store changes. Even though the behavior of the prototype is unexpected, it is quickly understood and liked by most participants. After all, it saves an additional click. For this reason we do not intend to change this behavior. To reduce the confusion it would be advisable to include some kind of feedback when the change is stored, like a transparent message that disappears after a while.

Similar items are indistinguishable

- Similar objects in the center and the history are indistinguishable. Similar to the related items, the most important neighbor should be displayed as well. This could be achieved by increasing the size of the history bar and dynamically managing that of the center.

The center area is too small

- The center area is too small. Some people would like to keep many shortcuts in the center to keep active items quickly accessible. We believe it makes more sense to integrate these shortcuts into the network itself, for instance by creating a favorites item, which can be accessed as easily as the home item and can be connected to any items of the network.

Navigation seems cumbersome

- Navigation seems cumbersome. The participant felt that he/she needed to do too many clicks to browse through the network. This is hard to comprehend without being able to discuss it with the participant. The issue should be considered more closely in future user tests.

Altering interactions are unclear

- Altering interactions like copying, creating associations, or deleting objects are unclear. This was a major problem of the evaluation and resulted in the failure of several tasks. It is hard to decide whether this is due to insufficient training or a general problem of the interactions. Supervised testing with more users and extensive training can settle this issue.

Time is poorly visualized

- Time is poorly visualized. Information highly dependent on time, like appointments, should be visualized on a time scale. I will discuss in chapter 9 how this could be realized in a future prototype.

The concept of associations takes getting used to

- The concept of associations takes getting used to. The network appears very confusing at first and users can easily get lost in it. Also, users tend to fall back to thinking in folders and hierarchies, which results in confusion and frustration. This is a problem of all radical innovations. The user is stuck in the metaphors of her current environment and unwilling to adapt to and learn new techniques, no matter how efficient they are. This can be countered by simulating certain aspects of the known environment and moving stepwise to the innovative system. For the Associative PDA, better visualizations of transitions and current state could improve this matter.

- Multiple pages are poorly visualized and the animation for browsing through the pages is distracting at best. I will discuss in chapter 9 how handling multiple pages can be improved by using techniques like the fish eye view.

Multiple pages are poorly visualized

- The display of each item's most important neighbor is useful, yet confusing, especially when the displayed neighbor is closer related to the current selection than to the item at hand. This problem is likely to be caused by the unfamiliarity of the artificial data set that was created for this evaluation. If the user creates the data herself, it is unlikely that connections of the network will confuse her.

The display of the most important neighbor is confusing

- Several participants have lost reference to a newly created item, resulting in the item being lost in the network. This problem could be principally avoided by automatically creating a reference in the center area when creating an item. Further, it is technically possible to identify such orphan items in the network and present them to the user.

Newly created items are lost

- Type items are indistinguishable from information items and the type cannot be accessed from the item. Some back reference to the type item should be included for all information items. This reference could be included in the list of related items or on a dedicated screen space. By providing this way back, the entire network can be considered as a hierarchy with the home item as the root item, which might help new users find their way.

Types are indistinguishable from items

Changes in the order of related items are confusing

- The order of the related items changes after accessing them because of a change of importance. This makes systematically going through several items a hard task and should be avoided. With a new visualization of the pages, as will be discussed in chapter 9, this problem should become obsolete. A back operation which takes the user back to exactly the same screen would also solve this problem.

More feedback is required

- More feedback is required for altering operations. This should be implemented in a future version.

The quantitative results were not statistically significant

The unattended evaluation of this prototype did not work out as well as we had hoped. The recording of task completion times did not yield any statistically significant results and the comments were partly hard to understand. Overall, an attended user study of fewer participants would probably have provided more reliable results.

An unattended evaluation does not work well with innovative systems

I would not recommend doing an unattended evaluation like this one with any innovative new systems at such an early stage in the design process. Many participants are scared off by time-consuming tutorials and lengthy explanations. Others skip the instructions and do not understand the interactions or work on the wrong tasks. All these problems are easily fixed by supporting the user during an attended evaluation but are virtually inevitable in a remote usability test. They are hard to identify and have great influence on the test results.

For the project, this evaluation has discovered several concrete flaws of the user interface, which should be fixed and retested in future prototypes. Further, it has shown that some users have major problems understanding some of the interactions and getting used to the concept of associations. It became obvious that the tutorial provided in this evaluation was no sufficient training for the test users and should be improved in future evaluations.

Chapter 8

Evaluation

"Interpretation is the revenge of the intellect upon art."

—Susan Sontag

As a final evaluation we have conduct another attended evaluation to compare the Flash prototype with a commercially available reference system.

A comparative study was conducted

The participants were first introduced thoroughly to both systems, their interactions, and the organization of information in the system. Then, participants were asked to solve four tasks on both systems. The time to perform each task was measured, and any conspicuities and comments were noted. The order of the system was alternated throughout the participants to even out learning effects.

Participants were made familiar with both systems

Afterwards, the participants were asked what system they preferred working with and whether they could imagine using a system like the Associative PDA for their daily work.

Similar to the paper evaluation, users were encouraged to describe what they were doing but not required to do so. Again, we decided to instead base our understanding of the underlying motivation on succeeding discussions of the observations.

Participants were encouraged to describe what they were doing

8.1 Participants

All participants were recruited from a computer science graduate course. Ages range from 20 to 26, and of the eleven participants, five are female. The participants are all students of varying majors: one is in media informatics, five in computer science, three in software-systems engineering, and two in technical communication.

The prototype was tested with experienced computer users

We wanted our participants to be very experienced computer users, to avoid technical challenges in handling either the test or the reference system. For a comparative analysis, it makes sense to test the interface with a uniform user population.

All participants were completely unfamiliar with the project and any of its prototypes. Some had experience with the reference system.

8.2 Set-Up

The evaluation was held in a controlled lab environment. Each participant was first introduced to the project and the general concept of associations. Afterwards, a complete introduction was given on the Associative PDA's and the reference system's interactions, content, and organization.

The Flash prototype was done on an Apple G5 running Mac OS X with a 23″ cinema display. The application was run on Safari with a simulated screen size of 350 by 482 pixels. To interact with the system an external mouse and keyboard was used. The content of the associative network was the same as for the Flash prototype (see chapter 7).

The reference system was a Pocket PC PDA

For the reference system an HP iPAQ 4150 PDA running Microsoft Windows Pocket PC 2003 was used. The device features a touch-sensitive screen with a resolution of 320 by 200 pixels. To interact with the system a pen and several dedicated hardware buttons to access common applications like the calendar, contacts, and the email inbox are used.

All of the information from the associative network was transferred to the reference system. To create a fair test environment, the organization of the information was converted to the hierarchical structure of the Pocket PC. The structure was distributed among three applications:

The associative network was manually converted to a hierarchy

- *File Explorer:* The file structure is accessed by the file explorer by selecting it from the start menu, located on the startup screen. It contains all course information and the seminar notes. At the root level there are two folders called 'Courses' and 'Notes'. In 'Courses', there is a folder for each course labeled with the title of the course. Inside, there is a folder for each year the course was held and another folder called 'Topics'. Inside the 'Topics' folders there are folders representing the topics relevant for the course. In each folder representing a year of a course, there is an excel spreadsheet containing a list of all students of the course with first name, last name, gender, and grade ordered by last name, and a word document called 'Course Outline'. Further, there is a folder called 'Evaluations' containing a text file named after each student of the course in the format 'Last Name, First Name.txt'. Under the 'Notes' folder there is a folder called 'Seminar', which contains a text file for each seminar named after its title. Figure 8.1 illustrates the structure of the file system visually.

  The file explorer shows a scrollable list of all files and folders of the active folder. A file or folder can be opened by tapping on it. The active folder is displayed above the list and by tapping on it, all superior folders are shown and can be selected. By holding the pen on an item for a second, a menu appears that lets the user delete an item or mark it for a copy or move operation. To finish the operation, the pen is held on the target folder and the paste command is selected from the appearing menu. This way of interaction is well-known to most users because of the copy-paste paradigm omnipresent in most current operating systems. Figure 8.2 shows a screenshot of the file explorer application.

- *Calendar:* The calendar is accessed by pressing a ded-

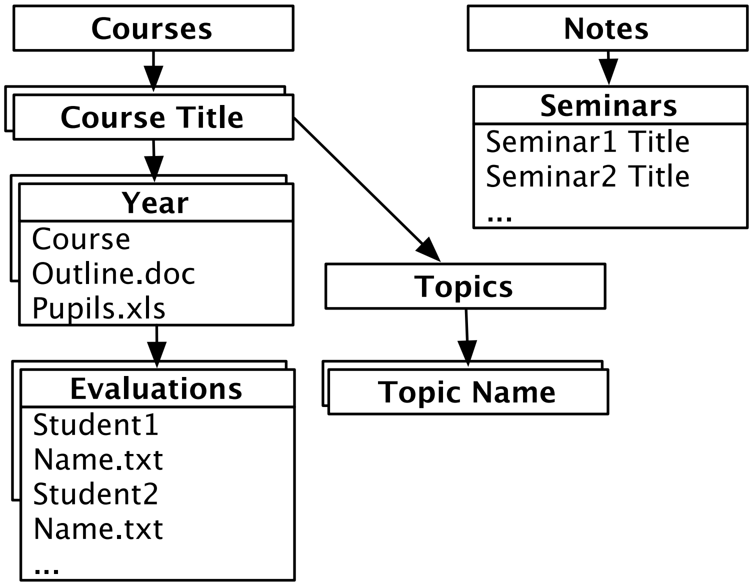

Figure 8.1: Organization of the file system on the Pocket PC

icated hardware key. It contains appointments at the corresponding dates for all seminars. These appointments are labeled by the name of the seminar, and the 'Teacher's Lounge' is entered as the place. The seminar notes are not stored with the appointments.

The calendar has five different modes of display: the agenda view shows the appointments of a day as a list of events; the daily view shows the appointments of a day by time; the weekly view shows the appointments of a week by time and weekday; the monthly view indicates days of a month with an appointment; and the yearly view indicates days of a year with an appointment. By clicking on an appointment, a detailed view is opened, where name, date, place, and other information about the event is displayed. Name and place are also visible in the agenda and the daily view. Figure 8.3 shows a screenshot of the calendar application.

- *Email Inbox:* The inbox is accessed by pressing another dedicated hardware key. It contains all messages from the associative network, with several messages being received on a single day. This was done to

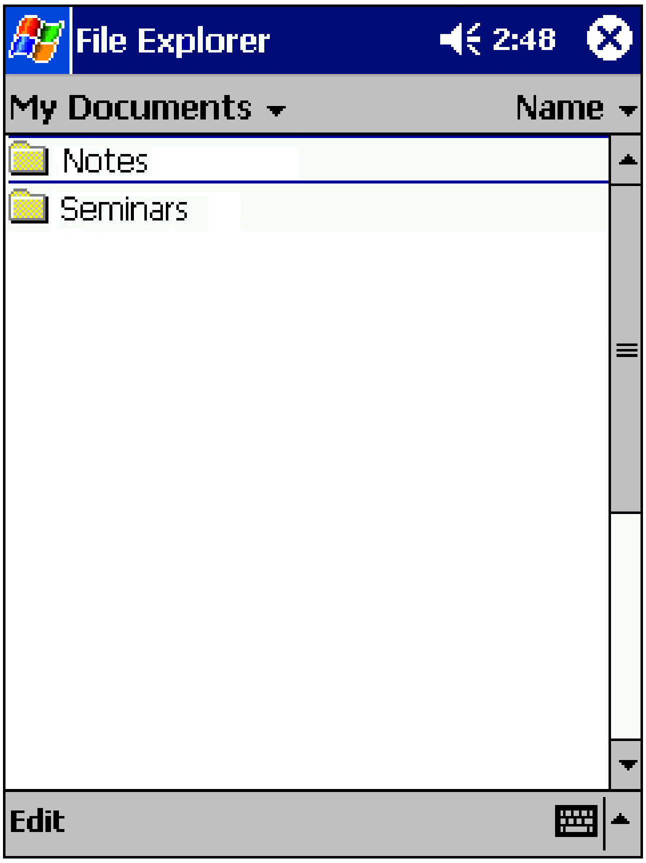

Figure 8.2: The file explorer application on the Pocket PC

simulate a normal work environment, where people receive many emails over a single day.

The inbox consists of a chronological, scrollable list, containing all received email messages. For each entry, the sender name, date, subject line, and size are displayed. Further, emails with an attachment are visualized with a special icon. Each item can be opened by tapping on the appropriate line. The detailed view then displays the message content on the main part of the screen and any attachments are listed at the bottom. These can be stored in the file system by holding the pen on them and selecting 'Save as..' from the menu. In the upcoming dialog a file name and the destination folder can be chosen. Here, only folders from the first level of the hierarchy, in our example

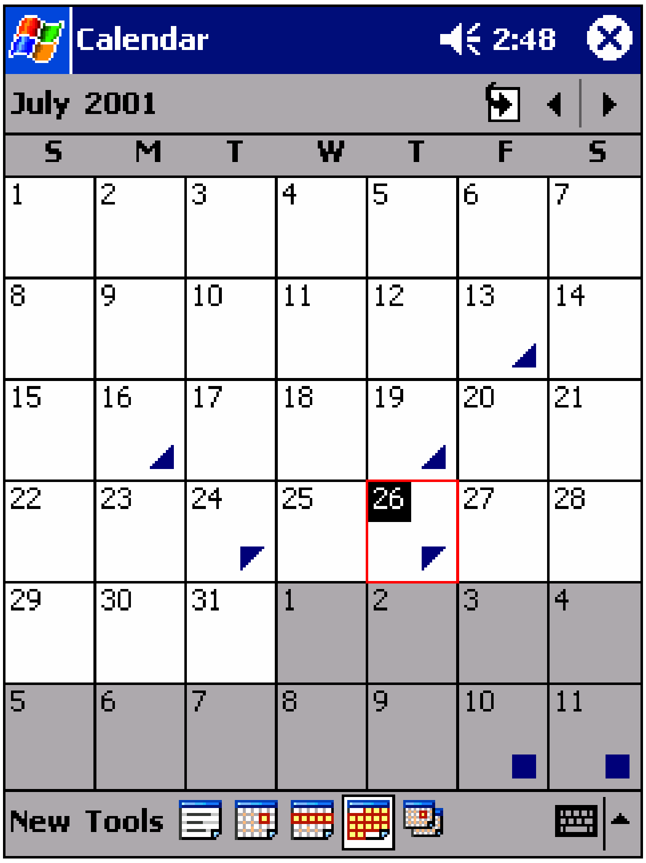

Figure 8.3: The calendar application on the Pocket PC

'Courses' or 'Notes' and several system folders, can be chosen. After storing the attachment this way, it is accessible through the file explorer and can be moved to another position.

8.3 Tasks

Each participant was given four tasks to solve with each system. The tasks were given in order with either the reference system or the prototype first. The order of the device was alternated to even out learning effects.

The tasks were chosen to be very similar between the systems

The tasks were chosen to be very similar but not identical between the systems. They had to be similar because we wanted to compare the performance of the systems for solving each task. On the other hand, they could not be identical, with the exception of task 1 and 4, because otherwise the participant would already know where the sought information was located on the second run of the task, which would be an unfair advantage.

When a user got stuck because he/she had forgotten a certain interaction or how the information was organized, the clock was stopped and the aspect was explained again. Afterwards, the clock was started again. This was done because we were interested in the performance of experienced users, who have knowledge of all the interactions and the organization.

Participants were assisted when they got stuck because of missing knowledge

The tasks for the Associative PDA were:

1. Find the grade and evaluation for Melissa Martinez, who is a student of English 2 2005.

2. Find the seminar notes, which have something to do with short stories, of a seminar held at the end of May in the Teacher's Lounge and connect them to the topic Short Stories.

3. Find the attached article of an email sent by Richard Ward and connect it to the topic World War I

4. Copy the course outline document from the course English 2 2005 to the course English 2 2006.

The tasks for the Pocket PC were:

1. Find the grade and evaluation for Melissa Martinez, who is a student of English 2 2005.

2. Find the seminar notes, which have something to do with writing, of a seminar held on a Saturday in August in the Teacher's Lounge and file them under the topic Analytical Writing, which belongs to the course English 2.

3. Find the attached short story of an email sent by Richard Ward and file it under the topic Short Stories.
4. Copy the course outline document from the course English 2 2005 to the course English 2 2006.

All tasks were explained in detail, and in addition, all interactions required to fulfill the task were reviewed to avoid later interruptions because of forgotten interactions.

8.4 Results

All of the participants enjoyed the prototype

All of the participants enjoyed working with the prototype and all would consider using it provided that it fit their daily work context and was polished more. Two participants especially liked the integration of the keyword search for attribute values, and one commended on the intuitiveness of the interactions for creating and deleting associations.

The majority of participants preferred working with the prototype

The majority of participants preferred the interactions of the Associative PDA over those of the Pocket PC. Two participants liked the operations in general better, while two others liked especially the drag and drop operations. One participant said that finding information on the Associative PDA seemed easier than on the Pocket PC, and another one was convinced that he/she performed better on the prototype, even though his/her average performance was better on the reference system. We guess this might have been caused by the user experiencing time passing by faster because of excitement and joy using the new system.

The Pocket PC outperformed the Associative PDA in three out of four tasks

Despite all this enthusiasm, the Pocket PC outperformed the Associative PDA in three of the four tasks by up to 44 seconds (220.0 %). The standard deviation values of task completion times are all fairly high (between 25 % and 50 %, with the exception of task 3 for the Pocket PC at about 15 %), indicating that the results are not overly reliable. The statistical measures are presented in figure 8.4.

The unreliability of the results is probably due to imprecise time measurements

Part of the unreliability of the results is due to imprecise

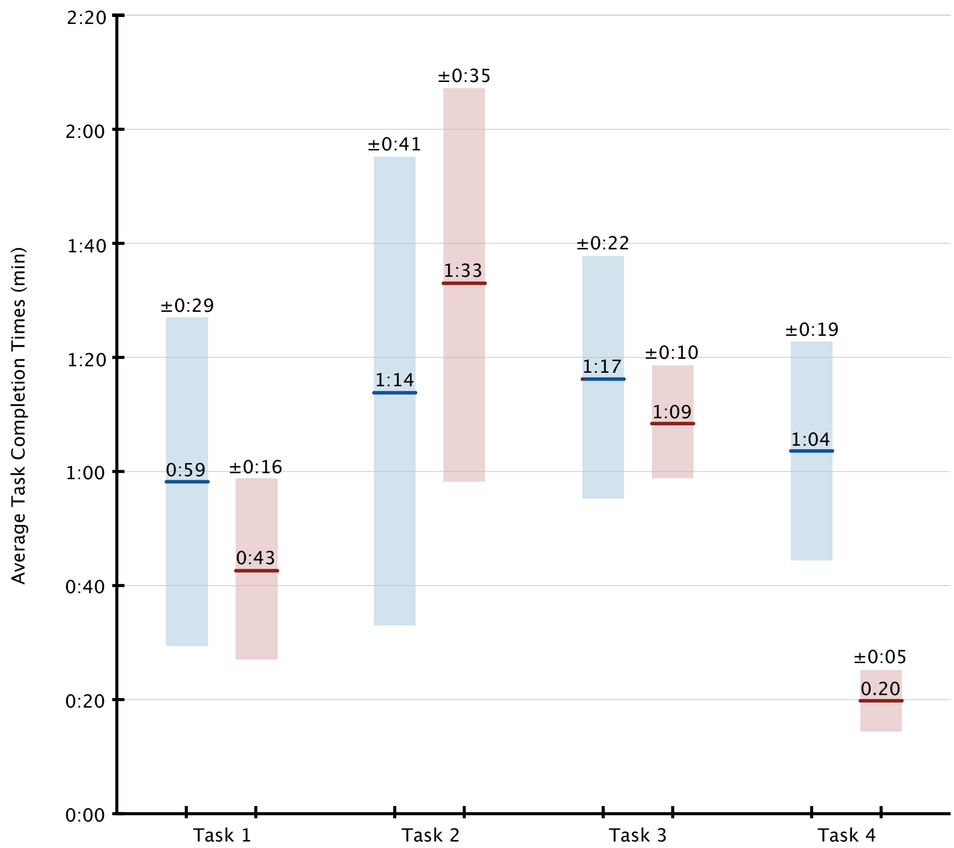

Figure 8.4: Average task completion times and their standard deviations of the Associative PDA in blue and the Pocket PC in red

time measurement and arbitrary breaks for explanations of the system's interactions. These were more common for the Associative PDA than for the Pocket PC, which explains why the time values for the latter are slightly more reliable. A general problem of this evaluation style is rooted in the personal nature of PIM and will be discussed in chapter 9.

A t-test showed that only for task 4 the two systems had significantly different means

Further, we performed a paired, one-sided Student's t-test to validate that the mean values for the two systems are significantly different from each other. The results show that this holds only for task 4 with $p < 0.05$. For the other tasks it can not be ruled out that the difference in their means happened by chance. Table 8.1 shows the result values of

The Associative PDA had a clear advantage in task 2

Some users had problems with the interactions of the prototype

the t-test for each task.

	t-test probability p
Task 1:	0.063
Task 2:	0.092
Task 3:	0.142
Task 4:	0.0002

Table 8.1: Results of the paired, one-sided t-test for each task

Three outsider values, more than twice the standard deviation above the average value, were ignored for the analysis.

The participants probably performed better on the Associative PDA for task 2 (assuming it was not by chance) because on the Pocket PC multiple applications had to be used to identify the correct information item. The users had to open the calendar application and find the seminar to determine its name before they could look for the notes in the file system. There is no way for the users to determine from the seminar notes whether the corresponding seminar was the one asked for in the task since no dates were listed with the notes. In the Associative PDA on the other hand, users could get an overview of all seminars and then identify the correct item by comparing the date attributes to the value given in the task.

This situation could be considered artificial, as users could store their notes in the calendar application or include a date in the seminar notes. Still, none of the participants objected against this separation when asked whether they would consider the organization as logical and construct a similar structure if they were in the suggested situation.

As in previous evaluations, the users had problems with some of the interactions and the data structure of the associative network. At several occasions, despite having explained the relevant interactions a second time right before the task, I had to interrupt the task evaluation and explain the interaction again. This had some impact on the accuracy of the measurements but not on the overall performance. It became apparent, though, that the interaction for creating a new object must be redesigned. I will discuss how this could be done in chapter 9.

The rest of this section concentrates on the user comments, which again yielded many interesting insights into what the users thought about the system. There were two concrete suggestions for improvement: If an information item has many attributes, a full screen mode should be implemented to view and edit all attributes at the same time, and the circles should be replaced by squares arranged around the center block to save space. The first suggestion is planned for a future prototype and will be discussed in chapter 9. The second suggestions is interesting and should be looked into more. The block arrangement impairs on the flexibility in the number of displayed items but saves a lot of screen space. I leave this for future research.

8.4.1 User Comments

The participants expressed several concerns about the prototype during the evaluation. For each such problem I will briefly discuss its consequences and how it could be solved.

The concept of associations is unfamiliar

- Three participants stated that the concept of associations is unfamiliar and takes getting used to. Another participant even claimed that he/she feels overwhelmed by the associative network. This was observed before and must be considered more in the future. One of the reasons for this problem is probably the unfamiliarity of the information. With their own personal information space, users are much less likely to feel so overwhelmed.

The center area is unfamiliar

- The concept of storing items temporarily in the center is unfamiliar and takes getting used to. The participant described the center area as a clipboard in an attempt to explain it with a metaphor from a familiar information environment. This problem is similar to the confusion arising from users not being able to drop the idea of files and folders while working with the system and is a general problem of innovative designs. Distinguishing the center area more from other UI elements might improve this matter.

Multiple ways to reach the information require a decision

- One participant criticized that he/she has to plan how to reach the information because there are multiple ways to reach it. This problem was found in the evaluation of paper prototype as well (see chapter 6). We believe that it is an artifact of the attended user study as the participant is eager to find the best way to the information and hope that this behavior will not happen in a more natural situation.

The associative network requires extensive maintenance

- Another participant stated that the efficiency of the associative network relies heavily on the number of associations and doubts that he/she would maintain this network thoroughly enough. This concern was expressed in the paper prototype as well (see chapter 6) and must be tested in a long-term study involving real user data.

Search items are indistinguishable

- The label of search items in the history is meaningless. In future prototypes, search items should be labeled to appropriately summarize their content.

Changes in the order of items result in confusion

- The position of important items, especially those related to the home item, should not change. This issue was also found and discussed in chapter 7. We hope that this problem will be fixed with a better visualization of multiple pages, which will be discussed in chapter 9.

Relying on a history instead of a hierarchy takes getting used to

- One participant considered the history as an alternative to the hierarchy. He/She stated that it makes sense but takes getting used to. This concern is related to users getting lost in the network because of its unfamiliarity. It might help especially new users if a hierarchy towards the home item was introduced, as discussed in chapter 7.

Scanning of items in a circle is harder than in a list

- Scanning of items in a circle layout is harder than scanning of lists. We do not believe this statement to be true because interacting with data arranged in a circle is very familiar because many modern clocks are arranged in a circle. It would be interesting to research this matter in a user test, but I will leave this for future research.

The appearance of the prototype must be overhauled

- One participant claimed that the icons are too small and do not make much sense. Another participant simply stated that the system is unaesthetic. A third participant said that he/she would consider using the system only if it was more polished visually. Conclusively, the appearance of the system seems to be a factor in its acceptance and should be improved in the next prototype.

- Similar objects are not distinguishable in the center and in the history. This problem was found and discussed in the evaluation of the Flash prototype (see chapter 7) and can be fixed by including the name of the most important neighbor in these items as well.

Similar objects are indistinguishable

This final evaluation confirmed several known findings of earlier evaluations but resulted in only few new insights about the system. The results of the comparative study are questionable at best because of the high variance and the failure of the t-test.

After the decent success of the last two evaluations, it has become apparent for us that there is a general problem of evaluating PIM systems, which will be discussed in the next section.

Chapter 9

Conclusion

"The significant problems we have cannot be solved with the thinking used to create them."

—Albert Einstein

My work on the Associative PDA has been a great experience because I could apply the knowledge I acquired during my studies of Human-Computer Interaction in a real design project and gather valuable practical skills in designing interactive systems and scientific research.

I started out by reading the current literature and getting to know the theoretical background, user studies, and proposed systems in the field of PIM. Afterwards, I have conducted a contextual inquiry to better understand current practices in the field. Then, in an iterative design process of three cycles, I have refined the initial design ideas by testing them with the users. Finally, I have evaluated and compared the final prototype with a commercial solution in another user test.

The project is not completed

This project is far from being completed, though. The state in which I leave this project, is a working and evaluated interactive prototype, simulating many aspects of how the final system will probably be like. I do not expect any major changes in the interface or its interactions over this version. Instead, future work should concentrate on implementing user-suggested features, transforming the prototype into a productive system, and evaluating it under more realistic circumstances.

There are still some open issues left from the various evaluations, which I did not find the time to implement and evaluate. I will discuss these issues briefly in this chapter and make suggestions on how they could be covered. Afterwards, I will explain why we believe that the disappointing results of the quantitative evaluations of the third prototype are caused by a general problem of evaluating PIM systems. Finally, I will finish the chapter by giving a summary of the most important points raised in this thesis.

9.1 Future Work

This chapter summarizes and discusses the issues found in the various evaluations, which I did not find the time to consider in my work. I will also provide ideas on how the issues could be addressed in future prototypes of the system.

9.1.1 Time

Improve the use of time in the organization

The use of time to organize information was left out of the prototype up this point because we were not decided on how to realize it. This is a deficiency though, since the contextual inquiry has shown that time holds valuable cues for information retrieval and the value of time has long been acknowledged in the literature. Some projects, like Lifestreams by Freeman [1997], go as far as using time as their primary attribute for organizing information.

Time is set on a linear scale and follows a strict hierarchy, where points in time are part of periods like a day, which in turn are part of longer periods like a month. Depending on the information item, a point in time or a period can be relevant.

In an associative network, time can be realized in two ways:

1. It can be stored as an attribute of the information. This way special interactions must be designed to quickly navigate to specific points in time or intervals. For instance, the view of time-related items could be realized as a time scale where items are represented by markers and the scale can be browsed and zoomed.
2. It can be stored as an information item that is connected to all items for which the time is relevant. This way the known interactions of browsing the network can be used for time as well.

In either case, the visualization of time-related information like appointments should be changed to reflect the course of time.

9.1.2 Authoring

Include a full screen authoring mode

So far, the prototype does not support any authoring of information content like the text of a document. Even though the system concentrates on the organization of information, it must be possible to at least view the content or finding information would be useless. It would also help to include simple editing functionality so the user does not have to switch to another system to perform even small changes on the content.

Authoring could be realized using a full screen edit mode, which uses the entire screen to show the content of an information item. This edit mode could be toggled using a dedicated hardware key at the side of the device. This mode makes sense, even though in general we want to follow the principle of modelessness proposed by Raskin [2000], because it makes good use of the severely limited screen space and its state is clearly visualized.

The proposed full screen edit mode would also make sense to display the attributes of items with very many attributes.

9.1.3 Integrated Search Results

Make search results accessible

It makes sense to keep search results for future reference. So far this can be done by storing them in the center or they can be found in the history. Both methods are not very practical, as the space in the center is very limited and should only considered as temporary, and browsing a linear history is not very efficient.

Instead, search results should be stored inside the associative network such that the user can organize them into her system. This can be realized in two ways: (1) the result list can be stored as a static list, which is defined when the search is performed or (2) the result list is stored as a dynamic list, which is redefined each time it is accessed by performing the search again. Both ways make sense and should be implemented. The latter must be specially visualized though to indicate its dynamic nature.

9.1.4 Item Importance

Improve the computation of item importance

So far item importance is computed by considering the number of times an item is accessed and the last time it was accessed. The reason why this is insufficient is best illustrated by an example: If a user is looking for the grade of a student in a specific course, one strategy is to identify the student, then move to the course, and expect the system to highlight the grade, because it is connected to both the current item and the last visited item. The current implementation does not act in this way.

To achieve the desired behavior, not only the time the item itself was last visited must be considered but also the time all of its neighbors were last visited. Items with recently visited neighbors should be given increased importance.

9.1.5 Mass-Operations

Support mass-operations

Support for mass-operations is currently very limited. Operations cannot be applied to multiple items at the same time. For large data sets this will probably become a major flaw.

Mass-operations could be supported by allowing to perform operations on all currently shown related items. This way the network and the keyword search can be used to define the target of the operation. An example of such a mass-operation is to connect all the students of a previous course to a new course to indicate that the entire class has advanced to this class.

9.1.6 System-Wide Keyword Search

Provide a system-wide keyword search

So far the keyword search was limited to the related items of the current selection. It makes sense to extend this search to the entire network because information items with unique attribute values like people can be found extremely quickly this way.

To preserve the local keyword search, results should be ordered by the distance from the current position. At first, direct neighbors of the current selection with matching attribute values are listed, then neighbors of neighbors, and so forth. This way users can use a combination of position in the network and keyword search to find information items, which might yield interesting strategies for storing and retrieving information.

9.1.7 Distance in Neighborhood Search

Extend the search of shared neighborhoods

In the final prototype the search of the shared neighborhoods of multiple information items could only find items directly related to all provided source items. This function should be extended to look through the entire network, ordering items by their distance to all provided items.

The ordering of items is influenced by the distance measure. If for instance the sum of the shortest paths from an item to each source is chosen, it does not matter whether the item is close to a single source or equidistant from all sources, as long as the paths add up to the same number. If on the other hand the sum of squared distances is chosen instead, items that are about the same distance from all sources are preferred over those very close to one source but farther away from the rest.

Choosing the right distance measure is important for the design of this function because if the user is only confused by the results, the function is worthless to her. Implementing several distance measures and letting the user choose between them is not a viable solution, because it is unlikely that the user knows which distance measure is best for her purpose. The most appropriate measure can probably be identified by conducting a user test. Alternatively, multiple measures could be implemented if visualized accordingly. For instance, close to a single source could be placed adjacent to the source or drawn in a similar shape or color.

9.1.8 Multiple Pages

Improve the handling of multiple pages

The evaluation of the final prototype has shown that multiple pages of related items are poorly visualized and hard to interact with. There are many visualization techniques available for information lists but they must be adapted to fit the circular design of the Associative PDA.

A fisheye view, for instance, could be arranged in a circle to visualize multiple items. The fisheye view shows the focused item larger than the rest, allowing the user to view its details while also providing an overview of the entire list. Furnas [1986] provides a formal definition of fisheye views and their application in information systems. One problem of this method is that it is hard to indicate focus on current touch-screens because touching the screen initiates a selection.

Another way to better visualize multiple pages of information is guided by a crank metaphor. The user winds up the information by turning a physical wheel at the side of the device. The circle of items is turned in accordance with the wheel to reveal more items, following the principle of natural mappings (see Norman [1988]).

For both interactions it is necessary to change the current implementation of the first page holding the most important items. On the fisheye view this special page could be removed entirely and important items could be indicated in the overview. For the crank the first page could be left as is but its end must be clearly indicated because the browsing does not happen page-wise any more.

9.1.9 Classification by Deduced Categories

Introduce adaptive classification

Categories for the classification of long item lists could be deduced from attribute values or associations. For instance, for people items the attribute gender is either male or female. If the user is presented with a long list of people, it might make sense for her to choose from these possible values for gender to narrow down the list.

Similarly, classification categories can be deduced from associations between the items. If for instance the user has organized her location items in several large items like country, which are connected to all contained items, the recall process could be improved by letting the user choose from these high-level items first and then from their related low-level items.

9.1.10 Versioning

Support versioning

Concurrent versioning systems like SubVersion[1] store every change of a document as a new version instead of overwriting the old document. This process avoids loss of information and allows the user to recover previous versions of the content that might hold valuable information.

This concept could be integrated into the Associative PDA as well. Earlier versions of information items could be realized as individual items connected to the next version.

[1] http://subversion.tigris.org

When storing a change, a new item is created and all the associations from the old version are moved to the new version. If visualized properly, it would not even be necessary to include version numbers because the order would be apparent from the network structure itself.

9.1.11 Custom Hierarchies

Allow the definition of custom hierarchies

By defining an item as a root item, the associative network can be used to simulate a hierarchy, with the depth of an item represented by its distance from the root item. This can be realized for any item of the network and makes especially sense for items that are logically organized in a hierarchy like locations or times.

It might also make sense to base such a hierarchy on the home item. This way users will probably be less likely to feel lost in the network because they can fall back on a familiar way of organizing information. To realize this, all information items have to be connected with their type item.

9.1.12 Item Creation

Improve the interaction for creating items

From the evaluation of the final prototype it became apparent that the interaction for creating copies of items is errorprone and should be redesigned. The interaction used for the paper prototype, even though more complex, turned out to be better because it allowed copying items directly into the correct context. Another way to improve the current implementation is to automatically create a reference to the new item in the center because it is likely that the users will want to connect it to other items.

We believe though that the interaction should be changed fundamentally. We observed many users struggling with it, loosing the copied item in the network or confusing it with the original. The interaction is likely to be so problematic because there is no equivalent of it in common systems. Here, items are either copied directly into the correct context or into the same context as the original. Maybe copying all associations of the original items might already improve matters.

This concludes the discussion of open design issues. A successful continuation of the project should consider and fix most of these issues. I will now discuss a general problem we encountered when evaluating the Associative PDA.

9.2 General Problem of Evaluating PIM Systems

Users know their personal information 'inside out'

Personal information management is based on the personal information of the user. This implies that the user has created and is actively working with the information. Conclusively, she knows the information 'inside out' and does not need to stop and think about where or how she stored something in her system.

Users do not feel familiar with an artificial data set

Our evaluation method was to generate an artificial data set, explain it to the users, and ask them to imagine it was their personal data. In reality the data is in no way personal to the test user and she will never identify with this data the way she would with her very own data. Therefore, any tests based on this artificial data will be biased by a missing understanding caused by unfamiliarity of the information.

PIM evaluations must be conducted on authentic personal data

In conclusion, a good user test in PIM must be based on authentic personal information. For an innovative system that overthrows some of the metaphors of current systems, like the Associative PDA, this implies that the user has to migrate all of her information to the new system and adapt its organization to the new environment. This process, though, requires an enormous amount of time and work, and if the new system fails, the test user is still stuck with it. A user test under these premises is unrealistic.

So, how can a PIM system be evaluated? To my best knowledge, current literature does not provide any guidance on this problem. The PIM systems with a released evaluation all integrate seamlessly into current systems, so their evaluation could be conducted by installing the system on the test users' systems and observing their new performance. This approach is not feasible for a system like the Associative PDA.

User tests on artificial data can reveal useful insights

The user tests we conducted did reveal many interesting insights, providing a good basis for an evaluation, but it is unclear how to continue. We could develop more prototypes and run the same tests with them, although I doubt that many interesting results will evolve from these evaluations. Somehow future prototypes must be tested under better premises with realistic data sets.

Reduce the scope of the user test to a subset of PIM activities

Maybe it would be possible to work with only a small subset of the information of a test user. The user could then spend some time on organizing the information using the new metaphors and getting used to the system. The tasks could then be adapted to the subset of information. This way the effort for creating a truly personal data set could be severely reduced. If this reduction is enough and the results remain relevant is questionable.

To conclude this chapter and the thesis, I will now summarize the most important points raised in this document.

9.3 Summary

The work on this project has been in the spirit of this chapter's quote by Albert Einstein. Current practice of PIM is bound by the metaphors predominant in modern operating systems. We wanted to break free from these limitations by dissociating from today's way of thinking about PIM.

We have achieved this goal and come up with an innovative system that breaks the accepted boundaries of PIM systems, for instance their inability to mirror every aspect of the user's way of thought. I believe, the Associative PDA is more flexible and logical than any other PIM system on the market or in the literature, without being too complex to be usable.

The qualitative user feedback of our evaluations suggests that my beliefs are not entirely far-fetched. People understand the system and appreciate the extra flexibility. They see the associative network as a logically concise way of organizing information.

The Associative PDA provides all the necessary interactions and visualizations to work with the associative network on a portable device. In the long run, the user will benefit from using it because of improved organization and retrieval of her information and the ability to access it ubiquitously.

Appendix A

Storyboards

Trying to find information that is not filed

Trying to find information that is not filed with the Associative PDA


Multi-Media-Madness
In a meeting...
yap yap yap yap yap
...in this study by Brown...
This is really interesting!
3 weeks later...
Do you know of a study we could include?
Uh, didn't we talk about something in a meeting a few weeks ago?
I am not sure, let me get back to you...
It's not in my notebook...
Emails? No... Project files? No... Maybe a system-wide search? But what keywords to use?
Maybe it's somewhere on my desk... Found it!
This should be all you need.

Multi-Media-Madness
with the Associative PDA

Incoming Task On-The-Go

Incoming Task On-The-Go with the Associative PDA

Bibliography

Deborah K. Barreau. Context as a factor in personal information management systems. *J. Am. Soc. Inf. Sci.*, 46(5):327–339, 1995. ISSN 0002-8231. doi: http://dx.doi.org/10.1002/(SICI)1097-4571(199506)46:5⟨327::AID-ASI4⟩3.0.CO;2-C.

Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. The user-subjective approach to personal information management systems. *J. Am. Soc. Inf. Sci. Technol.*, 54(9):872–878, 2003. ISSN 1532-2882. doi: http://dx.doi.org/10.1002/asi.10283.

Hugh Beyer and Karen Holtzblatt. *Contextual design: defining customer-centered systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 1-55680-411-1.

Richard Boardman and M. Angela Sasse. Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 583–590, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-702-8. doi: http://doi.acm.org/10.1145/985692.985766.

G. H. Bower. Analysis of a mnemonic device. *American Scientist*, 58:496–510, September 1970.

Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945. doi: http://doi.acm.org/10.1145/227181.227186.

Yuhan Cai, Xin Luna Dong, Alon Halevy, Jing Michelle Liu, and Jayant Madhavan. Personal information management with semex. In *SIGMOD '05: Proceedings of the 2005*

ACM SIGMOD international conference on Management of data, pages 921–923, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-060-4. doi: http://doi.acm.org/10.1145/1066157.1066289.

Steven Carmody, Walter Gross, Theodor H. Nelson, and Andries van Dam. A hypertext editing system for the 360. In *Pertinent Concepts in Computer Graphics*, Providence, Rhode Island, 1969.

B. Christie. *Shared Information Systems*. John Wiley & Sons, Inc., Chichester, UK, 1985.

Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale. *Human-computer interaction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997. ISBN 0-13-437211-5.

Peter F. Drucker. *Management: Tasks, Responsibilities, Practices*. Collins, 1973.

Eric Thomas Freeman. *The Lifestreams Software Architecture*. PhD thesis, Yale University, May 1997.

George W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM Press. ISBN 0-89791-180-6. doi: http://doi.acm.org/10.1145/22627.22342.

David Huynh, David Karger, and Dennis Quan. Haystack: A platform for creating, organizing and visualizing information using rdf. In *WWW2002 International Workshop on the Semantic Web*, volume 55, 2002.

Victor Kaptelinin. Umea: Translating interaction histories into project contexts. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 353–360, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-630-7. doi: http://doi.acm.org/10.1145/642611.642673.

Barbara Kwasnik. How a personal document's intended use or purpose affects its classification in an office. In *SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 207–210, New York, NY, USA, 1989. ACM Press. ISBN 0-89791-321-3. doi: http://doi.acm.org/10.1145/75334.75356.

Mark Lansdale. The psychology of personal information management. *Applied Ergonomics*, 19(1):55–66, 1988.

Thomas W. Malone. How do people organize their desks? implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, 1983. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/357423.357430.

Jakob Nielsen. Iterative user interface design. *IEEE Computer*, 26(11):32–41, November 1993. URL `http://www.useit.com/papers/iterative_design/`.

Donald Norman. *The Design of Everyday Things*. Basic Books, 1988.

Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design*. John Wiley & Sons, Inc., 2002.

Jeff Raskin. *The Humane Interface*. Addison-Wesley Professional, 1 edition, 2000.

Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997. ISBN 0201694972.

Carolyn Snyder. *Paper Prototyping*. Morgan Kaufmann Publishers Inc., 2003.

Sarah Waterson, James A. Landay, and Tara Matthews. In the lab and out in the wild: remote web usability testing for mobile devices. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 796–797, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-454-1. doi: http://doi.acm.org/10.1145/506443.506602.

Index

Typeset November 27, 2006