

myPIM: A Graphical Information Management System for Web Resources

Torsten Krohn
itemis AG
24118 Kiel, Germany
+49 431 5606 336
torsten.krohn@itemis.de

Martin C. Kindsmüller
Institute for
Multimedia and Interactive Systems
University of Lübeck
23538 Lübeck, Germany
+49 451 500 5175
mck@imis.uni-luebeck.de

Michael Herczeg
Institute for
Multimedia and Interactive Systems
University of Lübeck
23538 Lübeck, Germany
+49 451 500 5100
herczeg@imis.uni-luebeck.de

ABSTRACT

In this paper we present myPIM – a graphical information management system that provides user workflow requirements in research and teaching/learning context and the development of such a system. All users in this context are highly dependent on resources that can be found in the World Wide Web. The massive growth of online information continually increases in complexity, and needs to be managed efficiently and resourcefully.

Therefore, we developed a concept of a system for managing a user's bookmark archive, file archive, exchange of information with colleagues, support for finding and recovering resources, as well as optimizing the workflow in a research and teaching/learning context. This system is a web-based tool called myPIM (my Personal Information Manager) based on common Java and JavaScript frameworks.

A very important feature of myPIM is its dual use approach. Primarily designed as a personal knowledge management tool, this usage gives off information, which can support colleagues in discovering relevant information. As a collaborative or community tool myPIM tries on the one hand to employ the so-called "wisdom of the crowds" phenomenon, but on the other hand has to deal with the fact that the used ontologies are not fixed but co-evolve with their communities of use – an issue that was recently addressed in the new pragmatic web approach [27].

Categories and Subject Descriptors

H.4.1 [Information Systems Applications]: Office Automation – *Workflow management*. H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces – *Ergonomics, Graphical user interfaces (GUI), User-centered design*. H.5.3 [Group and Organization Interfaces]: User Interfaces – *Computer-supported cooperative work, Web-based interaction*.

General Terms

Management, Design.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Keywords

Human-Computer Interaction, Interactive Media, Information Management, Social Software, Social Bookmarking, Folksonomy, Tagging, World Wide Web, Internet-based Collaboration, Information Sharing, Visualization, Community, Awareness

1. INTRODUCTION

The development and growth of cheap broadband internet accesses for private households is well on its way. According to a survey of German public television and media companies [4] 62 % of all households have a personal computer with internet access. Ten years ago, only 6.5 % of the households had this access. Currently, over 90 % of all business companies and educational institutions are provided [31].

At the end of 2006, "You" was awarded "Person of the Year" by the Time Magazine [31]. Thereby, they singled out the new participation culture in the World Wide Web (WWW). So, it is not only consumption, but also through the contribution of content to weblogs, wikis or the like which defines the new digital lifestyle and adds to the rapid increase of information in the WWW. Google listed about 8.4 billion web pages [36], but Google's CEO Eric Schmidt explains: only 170 terabytes of the approximated 5 exabytes world's data is captured. It will take at least 300 years until all information is indexed and therefore is accessible for search queries [22].

Most users, who get in touch with internet resources, are more and more dependent on efficiently managing these resources. The amount of data increases exponentially, but the growth of individual knowledge and cumulatively the knowledge of groups is unknown. Fischermann [12] describes this situation referencing to history by saying that today's speed of getting information does not leave enough time to think.

This statement sounds as if it could be spoken at any media conference today, but in fact, it was Lord Melville, chief of British Admiralty, who said this back in the year 1860. This raises the supposition, that we can only handle this amount of information by providing structured information management and a sophisticated workflow. The majority (more than 92 % [4]) of the current one billion online users [7, 9] use bookmarks. Since internet users' interests are vastly diverse, we focused on developing a system for one specific group of users: researchers, educators, and students. But even such a narrow context is challenging, for we have to develop a system that encourages users to discover, understand, and even renegotiate the significance and definition of labels and labeling. In other words,

we have to offer a tool that supports the evolution of ontologies within communities as addressed in the pragmatic web approach [27, 28].

First of all, before we can design a concept of a system, that supports the requirements of its users comprehensively, we need to look at data, information, the user's knowledge and their workflow.

2. MANAGING INFORMATION

The economical and social systems of the Information Age are characterized such that the importance of information is seen as raw material or commodity. This became possible once the electronic data processing and global flow of information was established. The WWW is an important accelerator of this change in leaving the Industrial Age. However, the downside of this development is the fact that users sometimes get confused and not adequately informed because the WWW grows so rapidly, it becomes disorienting [37].

2.1 Data, Information and Knowledge

A crucial distinction must be made between data, information and knowledge. Information differs from data by having a context: for example, "25 degree Celsius" (data); "The temperature of this room is 25 degree Celsius." (information) [17]. Information is the result of an interpretation of data in regard to some existing or

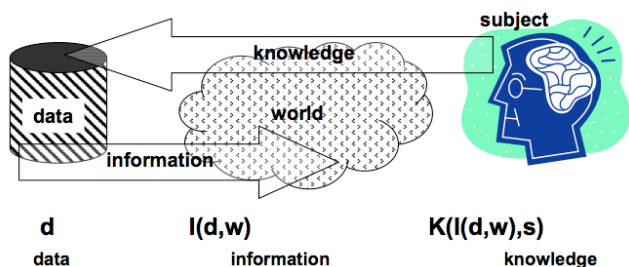


Figure 1. Data, information and knowledge. Data are entities whereas information and knowledge are relations [21].

imagined object or situation (Figure 1). Therefore, information can be modeled as a relation between data and a possible world [21]. Like information, knowledge does not exist on its own, but results from a cognitive process which represents information in an individual's memory. A formal description of knowledge would be a second order relation, namely a relation between information and a single individual.

Basic processes can be identified:

1. Interpreting: associating data to a (possible) world. This is done by an individual.
2. Representing: storing information in an individual's brain.

Taking this crucial distinction between data, information and knowledge into account, computer programs can process data and sometimes information but they cannot process knowledge as they do not have any. Nevertheless, computer programs can help individuals in managing their knowledge. Furthermore, knowledge is not the result of collecting information but comes

from successfully linking-up new information to prior knowledge.

Between the years 2000 and 2002 the amount of data doubled on the WWW. For the year 2010, it is predicted a doubling will occur every three days [14, 23]. It is therefore very likely that the amount of information will increase as well. But what about the development of individuals' knowledge?

Bookmarking is one method of dealing with data overload. Abrams, Baecker and Chignell [1] conclude that bookmarks are a simple tool for building personalized, highly relevant subsets of information where interesting or useful Web pages (URLs) can be stored for a later use. Users keep track of the resulting pointers to web pages by creating a bookmark archive — a personal web information space.

2.2 Searching, Recovering and Finding Something Similar

In the view of the complexity of information in the WWW, questions arise about how we can find useful information quickly and efficiently. When we have found what we searched for, how can we recover it at a later point in time? How can we find similar information about this topic?

Most archives are stored within a directory structure. If the number of bookmarks exceeds a certain limit (about 35 pieces [1]), the user makes categories and moves the corresponding URLs into the suiting ones. Archives with hundreds of bookmarks often have 20 categories or more. If an individual wants to remember an item (article, image, book reference), they have to go through a multi-stage process as can be seen in Figure 2: After the

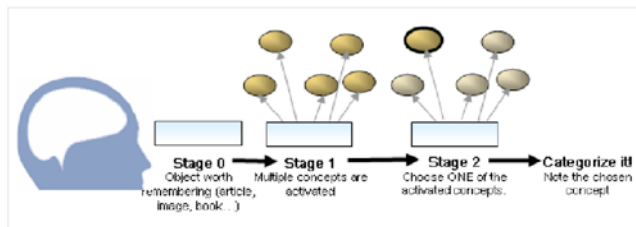


Figure 2. The underlying cognitive process for Categorizing [29].

individual makes a decision to remember something (stage 0), multiple concepts are usually activated (stage 1). As a last step, one of these concepts has to be chosen (stage 2). The information is then stored in the appropriate category. It might be difficult to access this information in the future, because the user has to remember what concept was chosen maybe months or years ago. So, this analysis interrupts the user's workflow. It is therefore essential to implement this action more efficiently.

A search in the WWW is always applied to certain topics, i.e. "at the moment" the user is searching for this specific information. The user's information needs are – as a rule – satisfied by browsing through their bookmark archive or by using search engines like Google or Yahoo. But frequently, the user likes to get more comparable sources. It seems reasonable to provide an opportunity so that they can benefit from the other peoples' knowledge. In a real world group of people, e. g. a business seminar, the participants of the seminar can share and communicate face-to-face. They know each other and their

respective field of work. This can be seen as filtered assistance by experts, which includes a semantic as well as a pragmatic analysis.

Nevertheless a conversation face-to-face or via e-mail is not always possible. An option is to try to make use of the “wisdom of the crowds phenomenon” in an online community. Surowiecki [33] describes in recourse to Galton [13]: “Under the right circumstances, groups are remarkably intelligent and are often smarter than the smartest people in them”. Surowiecki states four important factors that facilitate the “wisdom of the crowds” phenomenon:

1. diversity of opinion (based on diversity of information)
2. independence of judgments
3. aggregation of opinions
4. the fact that the group sets its own goal.

2.3 Workflow

With the distinction between data, information and knowledge stated above, it is possible to define the terms “Personal Information Manager” (PIM) and “Personal Knowledge Management” (PKM). The former is a (as a rule software) tool to help users organizing information. In PIMs several kinds of representations like addresses, notes, reminders and lists are saved and linked in appropriate and useful ways. Frequently the term PIM is used synonymous for applications like Microsoft Outlook or Lotus Organizer. A PIM can assist Personal Knowledge Management.

It is important to remember, that we often encounter information that we have not actively searched for at that particular moment. Instead, a lot of information is encountered in the course of time. Sometimes we come across interesting hints during work, which could be useful for some private interests. Then again, we find helpful articles for our job during surfing the web at home. Thus, the search results are still useful, but in a different context. Cutrell et al. [10] called these situations encountered information. This unexpected, but possibly useful information often interrupts the users’ current workflow. A PIM has to be able to manage encountered information in an efficient way.

It is possible to find multiple contexts by using certain tools. For example, somebody wants to call on a medical practitioner that a friend recommended a week ago. He does not remember the name or the telephone number of the practitioner, but, he remembers that he wrote it down somewhere; but where? In the address book (if yes, what was the name?), in the bookmark list (if he saved the link), in the e-mail inbox or did he move the e-mail already to a sub-folder? A useful PIM tool would deal with these multiple contexts e.g. by searching them automatically.

Especially in a research and teaching/learning contexts it is very common to use multiple different computers at the same time. Attached desktop computers at the office and at home, business laptops used in different locations on campus, private laptops used at home in combination with PDAs, smartphones, as well as other computers that are normally used by others like pool computers or computers for presentation at conferences.

3. RELATED WORK

Numerous projects in the past have addressed these problems. In the following chapters, we outline these ideas and describe the resulting software tools.

3.1 Early Systems

Wittenburgs approach of Group Asynchronous Browsing or GAB [35] suffers from the small overlap of the users’ interests. Marais and Bharat developed a tool called Vistabar [25] to support the users’ handling with web resources. It failed mainly because of the long-winded installation routine and the limited categorization options. Kanawati and Maleks [18] tried to classify bookmarks automatically with multiple agents. Most people rejected the system, since this automatism did not work properly. In the course of a NASA project Keller et al. created WebTagger [20] with the same purpose, but it failed because of insufficient compatibility with many web pages. Nevertheless, the basic research of the latter project has a tremendous standing.

3.1.1 Group Asynchronous Browsing (GAB, 1995)

The basic idea behind GAB is to use structural similarities in the users’ hierarchies of bookmarked URLs to offer new, relevant resources of the same topic for other users.

The procedure is illustrated in Figure 3. The capital letters in the graph represent URLs, the tree branches represents several users’

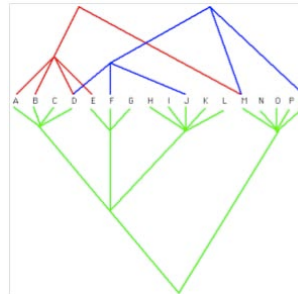


Figure 3. Three hierarchies in GAB [35].

bookmarks — one color per user. Bookmarked URLs are leaves in the branches. A node represents a topic which corresponds to its children. For instance, A, B, D and E are bookmarks of a special category of the red marked user. The green marked user has a stronger hierarchy in his bookmark archive.

URLs, which several users have bookmarked (for instance B green and red, D green, red and blue) are hyped by the system, so the corresponding web pages will rise in the hotlist.

It turns out that the overlap of the shared URLs is not big enough for the social component to be usable, because the critical mass¹ is never achieved. Yet, a calculation of the commonalities based on a larger field of interests could bear fruits. This could be an analysis of the user’s workflow by his e-mail and telephone activity.

3.1.2 Desktop Assistant Vistabar (1997)

Marais and Bharat’s Vistabar tool [25] observes browser activity. It is supposed to recover documents and supports the integration

¹ The critical mass is defined as the number of users in the network at the point when the number of users starts to grow exponentially [14].

of these documents into a so-called “shared knowledge store”. The Vistabar works as an assistant between the user and his browser essentially providing context awareness. Therefore, the tool generates a profile of the user's interests, looks after login and password inputs, exchanges system states with other users' assistants and analyses the contents of websites. Thus, it monitors the user activity and tries to automate frequently-used manual steps.

Some of the features of Vistabar were not implemented very well but the main reason for the failing of Vistabar was once again a critical mass issue: almost everybody wanted to use the feature “auto-bookmark” (suggestions for matching categories), but for this feature to work, many participants would have had to previously categorize these web pages.

3.1.3 WebTagger (1997)

Keller et al. [20] was developed by order of the NASA Ames Research Centers. It is a bookmark service for both personal managing and sharing of URLs within a defined group. According to the developers, the usage of classical bookmarks has many disadvantages:

- lack of facilities for sharing URLs
- inability to comparatively rank URLs according to their utility
- hierarchies are tedious navigational access models

They implemented a proxy application which inserts buttons (e.g. “categorize”, “retrieve”) directly into the top of web pages.

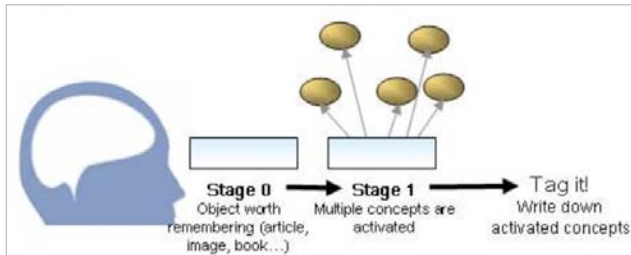


Figure 4. The underlying cognitive process for tagging [29].

After bookmarking, the user can set multiple categories, what would nowadays be described as tagging. The underlying cognitive process for tagging (Figure 4) differs from the process for categorizing (Figure 2). The user can recover the bookmark by choosing one of the concepts that was activated while tagging.

3.2 Current Systems

Presently, the range of web services available to users is remarkable. “Web 2.0” does not only provide ergonomic interfaces, but extends the users' scope. The user is consumer and information producer at the same time. Even though there are about 50 online bookmark managers available in the WWW, we limit our discussion to two dominant ones because all online bookmark managers have overlapping feature sets.

3.2.1 Dogear

IBM designed a tool called Dogear [11] to use the benefits of social bookmarking within a company. The tool shows the employees' names, and not their user names, to facilitate the process of forming communities of experts to a specific topic. Although the access is limited to the company's intranet, some

employees are scared-off sharing bookmarks because of this policy.

In Figure 5, a typical user's bookmark view is given. The tabs on the sidebar link to the individually used “tags” and a list of “people” who linked to the same page. The user can filter their tag cloud and control the number of tags shown. The more frequent a tag is used, the larger the word is displayed. The user can enter search terms and find matching bookmarks by means of tags or titles. The central area lists bookmarks according to their title, tags, description and date. A click on a tag shows all bookmarks tagged with this word. A click on the author's name provides all individual bookmarks.

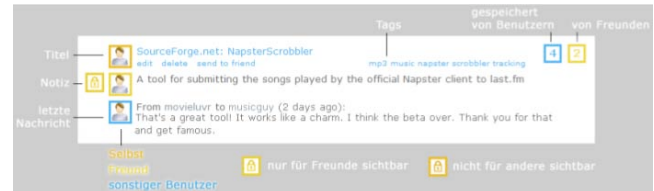


Figure 5. Representation of a bookmark.

3.2.2 del.icio.us

One of the first web-based bookmark managers was del.icio.us. In 2007, the service counted 1.5 million members. Similar to Dogear, the tool does not only store bookmarks, but helps the user to discover web resources of a selected information space. The time needed to bookmark web resources was highly reduced by means of a so-called bookmarklet². After that the URL is saved online and can be annotated with tags.

Some users like the plain interface, others prefer enhancements, e.g. better sorting possibilities, Boolean operators or better usability [15]. Moreover, due to the great growth of users, performance problems occur, especially on the weekends.

4. DESIGN

The most important lesson learned from the failures of the tools described above is that the users' needs and preferences are often neglected in the initial design of the tool. To avoid this oversight, we employ a so-called user-centered design process. The development process is oriented to the users and their tasks, purposes and properties to guarantee a high usability.

4.1 Use Cases

After interviewing potential researchers, educators, and students, we correlate the results to generate prototypical use cases (UC). A use case describes the interaction between a user and the system in respect to achieving a certain goal. This is the list of use cases for the system:

UC1: Keep bookmarks in mind under time pressure

UC2: Have fast access to frequently used bookmarks

UC3: Research new resources³ for lectures

² A small JavaScript program that is usually stored in the browser's bookmarks list or toolbar.

³ The term “resource” indicates that the intended system should support more than just URLs.

UC4: Store researched resources

UC5: Retrieve resources

UC6: Find similar resources

UC7: Handle tasks and encountered information

UC8: Forward information to groups

UC9: Insure security and avoid barriers

UC10: Handle bookmarks graphically (drag & drop)

UC11: Migrate from a previous to the new system.

In the use cases UC4, UC5, UC6, UC8, and UC10 it is especially relevant that users can discover, understand and maybe even renegotiate the meaning of certain labels and labeling schemata that are used by oneself and by other users.

4.2 Features

Based on these use cases, we derive features (F) for the planned myPIM tool. Some of these features must be “pragmatically enhanced”, in the sense that they have to take into account or facilitate ontology evolution and/or ontology negotiation [27]. These features are marked with PF. Referring to the everyday workflow's efficiency, from use cases UC1 and UC2 we can derive the following features:

F1: Tag-based system (Folksonomy [34]). We start with a tag-based system and can evolve into a more complex representation if the pragmatic ontology building process of the community demands it (PF).

F2: Recommending changed titles: A web page title is not always meaningful. If another user changed that title, it could be useful to suggest it in the bookmarking process (PF).

F3: One-Click-Bookmarking: Do bookmarking in a “Getting Things Done” (GTD) manner.

F4: Getting reminders of work to do, e. g. bookmarks to organize or to read.

Referring to finding and recovering resources we can derive following features for use cases UC3 to UC6:

F5: Sorting bookmarks with one click.

F6: Displaying frequently and recently used bookmarks.

F7: Favorites: Possibility to manually ‘pin’ bookmarks to a fast accessible location.

F8: Show the users “behind” the bookmarks, e. g. linking a personal blog or curriculum (PF).

F9: Displaying link- or blogrolls (PF).

F10: Subscribe to user's bookmarks to get notifications about new ones (PF).

F11: Mirror bookmarked web pages

F12: Notifications about change or loss of bookmarked resources

F13: Notifications about orphaned bookmarks (never or rarely used)

F14: Discover and show renamed online resources (PF).

F15: Typing of resources (literature, images, videos etc.) (PF).

F16: Store bibliographical references in a repository, including automatic attributing, cover image, bibtex export etc. (PF).

F17: Indicate in browser if already bookmarked.

F18: Auto-Completion for bookmark infixes.

F19: Full-text search in bookmarked contents.

F20: Search by author's name (for and bibliographical references and websites).

F21: Boolean search (AND, OR etc.).

F22: Search in bookmarks of a certain workgroup (PF).

From all further use cases (UC7 to UC11) we can derive:

F23: Assign tasks for bookmarks.

F24: Share resources with a links-for-you function (PF).

F25: Define access rights for notes (PF).

F26: Check URLs against constantly updated blacklists of suspected phishing and malware pages.

F27: Customize user preferences (search everywhere or just in titles, open history for all or limit access to it, etc.) (PF).

F28: Display thumbnails of bookmarked web pages.

F29: Visually arrange bookmarks like post-its to explore ideas (PF).

F30: Bulk import of bookmark archives (ZIP, HTML links etc.).

All features were grouped into feature sets. Each set consists of related features, i.e. all features of a set have to achieve a common purpose. These purposes are subject to dependencies, since they are built on one another. One easy analogy to describe this dependency is car manufacturing: you cannot paint a car, if the chassis is not already finished.

5. MY PERSONAL INFORMATION MANAGER

The myPIM tool consists of a core system (chapter 5.1 to 5.3) that contains the main functionality like entering, displaying, retrieving and distributing information. All further features (chapter 6.2) dock to the core system. As we employ a feature-driven development process [8] as our software engineering paradigm, we are able to deploy the basic features of the core system quickly, without negative effect on the expandability of the system later on.

5.1 User Interface and Navigation

Web pages are bookmarked to be displayed in browsers. Moreover, most users are familiar with handling a browser, so we build the tool with a typical client-server architecture accessible with all up-to-date browsers on all relevant operating systems.

Our interviews revealed that users bookmark resources in their browsers, take notes on sheets and mail links to other users and even to themselves. myPIM is supposed to merge these activities into a single context. As the user interface is the main frame of

activity it should be implemented first. The computer-based solutions for PKM, namely a graphical PIM, a social bookmarking service, and blog comments to communicate with each other have to be integrated into one tool. As Carrol, Mack and Kellog [6] show, choosing the right metaphors can considerably enhance the understanding of a system. We therefore decided to choose metaphorical names for the menu items, which relate myPIM's functions to real world tools that perform similar functions. Matching icons were added next to these terms to enhance this effect.

There are four main areas: dashboard, bookmarks, community and login/signup.

At the dashboard, the user finds logs about the latest activities of his community's friends as well as of himself. Moreover all projects and the newest tasks, that should be completed, can be seen here. As summarized by Sinha [29], tags are good as fast accessible pointers to your knowledge, but bad for organizing, restructuring and working with your knowledge. To countervail that, we provided some basic project management functions for entities like bookmarks, notes and tasks. Figure 6 shows the very first paper mockup.

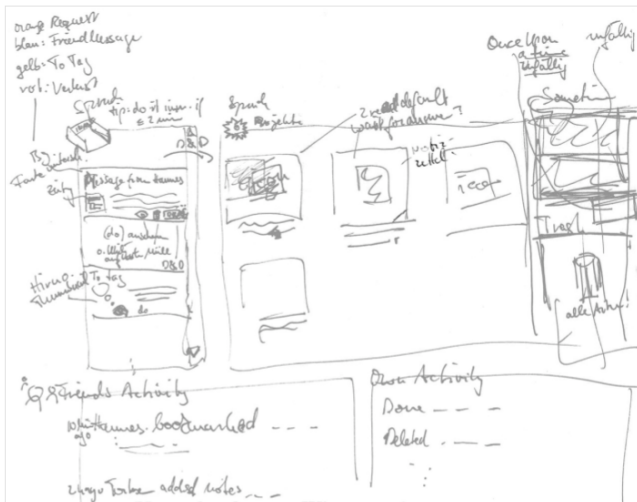


Figure 6. Paper mockup of the dashboard.

The bookmark area is for searching his own and others bookmarks. Also inserting, editing and deleting of bookmarks are performed here. The user can customize the view (title, tags, URL, user icon, sort). To support the community awareness, the latest comment regarding a certain bookmark is displayed in the large view.

The community area provides the standard functionality for maintaining relationships and tasks, i. e. friend requests, 'breaking up friendships' as well as the list of conducted conversations.

A very important design goal was that starting to work with myPIM should be made as easy as possible. Therefore we designed the signup and setup procedure as a three-step process displayed on a single page. After initially working with paper mockups, we designed a screen prototype of the signup/login page to get a first impression of the look and feel (Figure 7).

For the presentation and control layer we choose the Apache Struts framework. Thus, we obtain the separation of model, view

and controller and can make use of included functions. Moreover the integrated template engine Tiles ensures the paradigm of non-doubled code [16]. Many actions can be performed without (re-) loading a whole webpage (Figure 10). This asynchronous data transfer between browser and server is done with the help of the Java-Script frameworks Ajax, Prototype and script.aculo.us.

5.2 Bookmarking and Workflow

As stated in paragraph 5.1 it is important to manage all relevant information in only one context. At this we are geared to the time- and self-managing method called "Getting Things Done" [3]. This method encourages writing down upcoming activities to unburden one's mind. So, we do not have to remember everything that needs to be done and can therefore concentrate on actually performing those tasks. With regard to the tasks that our system is designed to support these activities deal with URLs, notes and news from other users. myPIM provides context-related to-do lists so that the user does not have to worry about forgetting something.



Figure 7. Screen mockup of the signup dialog.

Allen [3] describes the core process in five stages: We

1. *collect* things that command our attention;
2. *process* what they mean and what to do about them;
3. *organize* the results, which we
4. *review* as options for what we choose to
5. *do*.

Stage 1 is supported by listing the news and tasks. So, the user can plan his actions (Stage 2) by overlooking the summary and detail view. We organize the results in stage 3 with our project management tool or complete them immediately. myPIM supports the user with many drag & drop operations to do this like filing something in the physical world. Allen recommends looking over the projects in periodical cycles. Doing so, the user can take notes

(Stage 4) which helps to do the next activities (Stage 5). While Allen discriminates between references and trash (as project folders), Mann [24] proposes to unite these folders to get rid of the cognitive process in that the user has to decide where to move the object, since usually it is not known if this information is still in need. By default myPIM acts on Mann's suggestion [24] but can be adjusted by the user to support Allen's suggestion [3] instead.

Figure 8. New bookmark dialog.

To save URLs and tags in a database, we have to do some preliminary considerations. It is obvious to pay attention to canonical forms to minimize redundancies and to prevent inconsistency which can arise by inserts, updates or deletions. Our database schema (without tags) therefore complies with Boyce-Codd form, so it is stricter than 3rd normal form.

The usage of tags often creates performance weakness. How to map tags in a relational database is debated a lot. Generally three approaches are state of the art [19]: MySQLicious (one normalized table), Scuttle (two tables with one index) and Toxi (three tables with surrogate keys). Pipes [19] recommends five tables (one main table, four mapping tables) to support tag-typical⁴ inquiries “from left” with primary key, “from right” with index. Thus, we substitute hard disc memory by performance. Moreover we use the Hibernate framework [5] as persistence layer and for object relational mapping between Java objects and relations in a SQL-DBMS. Therefore we expect the system to scale up even if the community gets very large.

Figure 9. Bookmark search dialog.

⁴ For instance tag clouds, “all posts to one tag”, “all tags to one post”.

The appearance of a bookmark in the user interface is shown in Figure 5. To facilitate the community building process, all information is accompanied with a user icon on left. In particular, we added the latest message about this bookmark at the end. This generates awareness of the active users and the latest activities related to a certain resource and allows for fast answering of

Figure 10. Dialogs after asynchronous data transfer.

questions and commenting. The dashboard shows a list of all incoming and outgoing comments to the user's bookmarks. On the top, next to the title the standard functions are located: editing bookmarks, deleting bookmarks, and sending messages. Right aside is a list of all tags and the number of users and friends who bookmarked this link. In the middle the user can find notes (his

Figure 11. Community dialog.

own as well as other community members' public notes). On the very left lock-icons indicate whether the access is limited to the user's friends or only himself. We use a common color coding scheme for all modules of myPIM: blue for public, yellow for friends, orange for private access only. The bookmark view is especially designed to satisfy two important principles established by Ågerfalk and Sjöström [2]: *identity cultivation* and *maintained intentionality*. The first addresses the issue that a web artifact like myPIM as a rule acts as an arena for identity building activities. The latter one refers to the maintenance of intentional awareness, i.e. the users should be supported in staying aware of the consequences of their activities.

To bookmark a resource, the user gets the dialog in Figure 8 via a bookmarklet (see below) or via a simple hyperlink on myPIM. Title and URL is filled in automatically, some tags are suggested, if someone has bookmarked and tagged this resource before.

5.3 Searching and Community

To search for a bookmark, the user enters the search terms into the input field (Figure 12), checks the desired search parameters (tags,

title, notes, URL), specifies the scope (own, friends', others' bookmarks) and hits return or clicks the button. The results are displayed in a sort order, which can be specified in the upper right corner. For a fast search without visiting the myPIM web service first, the user can add a searchlet to his browser (cf. Figure 9). This is usually done during the sign-up process.

No matter which way of accessing the myPIM search function was used the search always returns the same kind of results page (cf. Figure 12). So, all displays of bookmarks are interlaced with themselves and with the search.

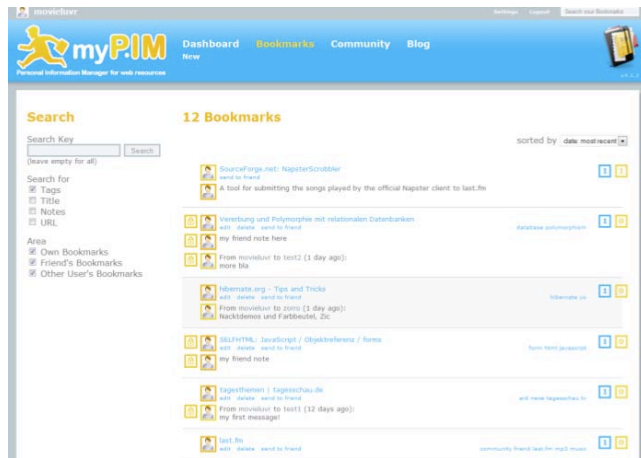


Figure 12. Bookmark search and results.

A click on the user's icon opens a dialog to start a 'friend request'. The respondent gets an e-mail about this request and can confirm or decline a 'relationship' with the requester. The community overview screen displays all 'friends' (Figure 11) and offers means to manage the 'relationships'. In addition to that the friends' and own activities are listed here to further improve the community awareness.

6. CONCLUSION AND FUTURE WORK

We presented a detailed description of our concept and the core features of our system myPIM. In the remaining part of this paper we will introduce further features that are part of the advanced system. These features are implemented as modules, which can plug into the core system.

6.1 Types and Views of Resources

Up to now, the search results are displayed on a new page, which has to be loaded. This results in an unproductive loading time and requires the user to press a search button. A solution with higher usability is possible by asynchronous handling of the data traffic, i.e. while entering the search term the system performs the search asynchronously and incrementally. The results are therefore listed on-the-fly and without a page load.

Resources are manifold, so they could be distinguished by their types (PDF, web page, image etc.) and sorted and grouped accordingly. The document types can be indicated by icons or even thumbnail representations of the documents.

6.2 Import

The migration from previous (browser-based or online) bookmark archives to myPIM can be performed with an import function. All

up-to-date browsers and online bookmarking services can export their bookmarks to a text file according to the Netscape standard. This file can be imported into the system, extracted and added to the existing bookmarks. If desired, the user can tag each bookmark or create a reminder to do this later.

Most users have a deep bookmark hierarchy, so we can use this structure to suggest some tags automatically. For example,

old hierarchy: „Knowledge Management“ > „Article“ > „Processing - eLearning Research Center“

new tags: „knowledgemanagement“, „article“

name of bookmark: „Processing - eLearning Research Center“

6.3 Advanced Search

So far, the search function uses a single string. The advanced search will use the Boolean operators AND, OR, and NOT as well as brackets and wildcards. Furthermore, the user can restrict the results set to one or several document types and can specify whether the search term should be found anywhere or only in the title, comments, tags, access date, full text etc.

6.4 Mirroring and Indexing

To combine bookmarks and a file archive, we need a repository. During the bookmarking process, the current web page is saved in the repository and therefore always available. There will be no need any more to manually backup web resources. Additionally, the user could snapshot the web page to freeze a certain state (versioning).

Beyond this, the system can be aware of losing the online contents of a bookmarked web page. The user can be informed by e-mail, newsfeed or a message on the dashboard about losses. To implement this feature, it is necessary to separate the content the user is interested in from secondary information. Reinke [26] developed a tool that performs this separation. Using this approach, we store only the relevant information (so-called pruned version of a web page)

By indexing the web pages, myPIM provides a full-text search through all resources. Again the pruned page is used as the search space, so only relevant information is searched.

6.5 Literature

Nowadays many books and journal articles are also available as PDF files, an electronic version of a paper in scientific research is now deemed to be standard case. Therefore, we establish a further data type 'literature'. Of course indexing and mirroring is reasonable here too. So, myPIM could render bibliographic management software obsolete.

6.6 Visual Organizing

Users of the current myPIM system have no further need of physical notes, that lie around on the desktop or frame their screens, but the familiar haptic feel of notes is gone with only textual references. Provision is made for a graphical representation of bookmarks as notes in an e.g. post-it style as compensation. Especially for literature items, this feature is useful, because most scientists and students already organize their research in this manner.

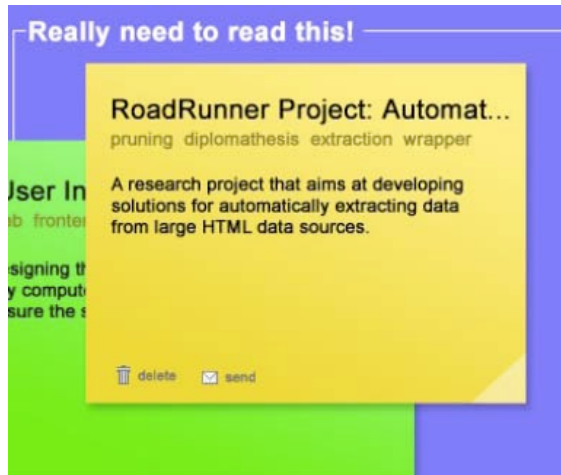


Figure 13. Community dialog.

The bookmarks within a project can look and be arranged like in Figure 13. In this example, the user arranges his references in the areas “read and useful”, “definitely read” and “probably read”.

6.7 Enhanced Pragmatic Web Support

We have demonstrated how considerations regarding the pragmatic web approach and the communication action perspective on web artifacts can be employed to enhance our myPIM system. This system is a prototypical representative of a new kind of tool that strives for supporting the pragmatic duality [30] by constituting an artifact for managing the user’s personal information space and at the same time communicating to other people through the artifact.

7. ACKNOWLEDGMENTS

Special thanks are due to all interviewees. Without your contributions, hints and ideas none of this would have been possible. We also like to thank Lia Hadley (University of Lübeck) and Hannes Schulz (Dräger AG, Lübeck). Special thanks also to Monika Krohn, Katrin Patermann and Tobias Kochems.

8. REFERENCES

- [1] Abrams, D., Baecker, R. and Chignell, M. 1998. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. CHI, Los Angeles. 41-48.
- [2] Ågerfalk, P. J. & Sjöström, J. (2007). Sowing the seeds of self: a socio-pragmatic penetration of the web artefact. In Proceedings of the 2nd international Conference on Pragmatic Web (Tilburg, The Netherlands, October 22 - 23, 2007). ICPW '07, vol. 280. ACM, New York, NY, 1-8. DOI= <http://doi.acm.org/10.1145/1324237.1324238>
- [3] Allen, D. 2003. Getting Things Done. The Art of Stress-Free Productivity. London, UK: Penguin
- [4] ARD/ZDF-Onlinestudie 2007. Über 40 Millionen Deutsche im Netz. ZDF Pressestelle, Hessischer Rundfunk. (Aug. 2007), <http://www.ard-zdf-onlinestudie.de/index.php?id=68>, (accessed on May 2, 2008).
- [5] Bauer, C., & King, G. (2006). Java Persistence with Hibernate. Greenwich, CT: Manning Publications.
- [6] Carrol, J.M., Mack, R.L. & Kellog W.A. 1988. Interface Metaphors and User Interface Design. In Helander, M. (Ed.). Handbook of Human Computer Interaction. Amsterdam: Elsevier, 67-85.
- [7] CIA World Factbook. 2007, Rank Order Internet users. <https://www.cia.gov/library/publications/the-worldfactbook/rankorder/2153rank.html>, (accessed on May 2, 2008).
- [8] Coad, P., Lefebvre, E. & De Luca, J. (1999). Java Modeling in Color With UML: Enterprise Components and Process. Upper Saddle River: Prentice Hall.
- [9] Computer Industry Almanac. 2007. <http://www.c-i-a.com/pr0106.htm>, (accessed on May 2, 2008).
- [10] Cutrell, E., Dumais, S. & Teevan, J. 2006. Searching to eliminate personal information management. Communications of the ACM archive. Volume 49, Issue 1. 58-64.
- [11] Feinberg, J. & Millen, D. R. (2005). IBM Watson Research Center. Project Dogear. <http://domino.watson.ibm.com/bridge/research.nsf/242252765710c19485256979004d289c/1c181ee5fbcf59fb852570fc0052ad75>, (accessed on May 2, 2008).
- [12] Fischermann, T. 2003. Next Economy. Der zweite Anlauf auf die Internet-Revolution. Berliner Taschenbuch Verlag, Berlin.
- [13] Galton, F. 1907. Vox populi. Nature, No. 1949, Volume 75, 7.3.1907, 450-451.
- [14] Gilder, G. 2003. Telecosm: The World After Bandwidth Abundance. Free Press. New York, NY.
- [15] Hood, S. 2007. Usability Lab. del.icio.us Blog, <http://blog.del.icio.us/blog/2007/07/usability-lab.html>, (accessed on May 2, 2008).
- [16] Hunt, A., Thomas, D. 2000. The Pragmatic Programmer. Addison-Wesley Longman, Amsterdam.
- [17] Kamschilow, M. M. 1977. Das Leben auf der Erde – Evolution der Biosphäre. Deutsch Harri. Leipzig, Jena, Berlin.
- [18] Kanawati, R. & Malek, M. 2002. A multi-agent system for collaborative bookmarking. In Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS). Bologna, Italien: ACM Press. 1137-1138.
- [19] Keller, P. 2005. Tagsystems: Performance Tests. Then each went to his own home. Philipp Keller's thoughts on tags and other sweets. <http://www.pui.ch/phred/archives/2005/06/tagsystems-performance-tests.html>, (accessed on May 2, 2008).
- [20] Keller, R. M., Wolfe, S. R., Chen, J. R., Rabinowitz, J. L. & Mathe, N. 1997. A bookmarking service for organizing and sharing URLs. Computer Networks and ISDN Systems 29 (September). 1103-1114.
- [21] Kripke, S. 1963. Semantical analysis of modal logic. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 9:67-96.
- [22] Kuri, J. 2005. Google will in 300 Jahren am Ziel sein. Heise online News. C't – Das Magazin für Computer und Technik, (Oct. 10, 2005). <http://www.heise.de/newsticker/meldung/64749>, (accessed on May 2, 2008).

- [23] Lyman, P., Varian, H. R., Swearingen, K., Charles, P., Good N., Jordan, L. L. & Pal, J. 2003. How much information? University of California, Schools of Information Management & Systems, Berkeley.
- [24] Mann, M. 2007. Inbox Zero. Google Tech Talk. <http://video.google.com/videoplay?docid=973149761529535925&hl=en>, (accessed on May 2, 2008).
- [25] Marais, H. & Bharat, K. 1997. Supporting cooperative and personal surfing with a desktop assistant. In Proceedings of the Tenth Annual ACM Symposium on User Interface Software and Technology, Banff, Alberta, Canada (October 14-17). New York, NY: ACM Press. 129-138.
- [26] Reinke, C. 2007. Optimierung von Reload-Strategien im Web durch Komponenten-Tracking. Institute for Information systems, University of Lübeck.
- [27] Schoop, M., Moor, A. d., & Dietz, J. L. 2006. The pragmatic web: a manifesto. *Commun. ACM* 49, 5 (May. 2006), 75-76. DOI= <http://doi.acm.org/10.1145/1125944.1125979>
- [28] Singh, M.P. The Pragmatic Web: Preliminary thoughts. In Proceedings of the NSF-OntoWeb Workshop on Database and Information Systems Research for Semantic Web and Enterprises, (Apr. 2002), 82-90.
- [29] Sinha, R. 2005. A cognitive analysis of tagging (or how the lower cognitive cost of tagging makes it popular). Thoughts on technology, design & cognition. http://www.rashmishaha.com/archives/05_09/tagging-cognitive.html, (accessed on May 2, 2008).
- [30] Sjöström, J. & Goldkuhl, G. (2004) "The Semiotics of User Interfaces: A Socio-Pragmatic Perspective," In *Virtual, Distributed and Flexible Organisations: Studies in Organisational Semiotics*, Liu, K. (eds.), Kluwer Academic Publishers, Netherlands.
- [31] Statistisches Bundesamt 2007. Nutzung von Informationstechnologie in Unternehmen – Ergebnisse für das Jahr 2006. Statis, Wiesbaden. 32-34.
- [32] Stengel, R. and Grossman, L. 2006. You – Yes, You – Are TIME's Person of the Year. *TIME Magazine*, 26/2006. 28-29.
- [33] Surowiecki, J. 2005. *The Wisdom of Crowds*. Random House Inc. New York, NY.
- [34] Vander Wal, T. (2007). Folksonomy Coinage and Definition. 02.02.2007. <http://www.vanderwal.net/folksonomy.html>, (accessed on May 2, 2008).
- [35] Wittenburg, K., Das, D., Hill, W. C. & Stead, L. 1995. Group asynchronous browsing on the World Wide Web. Proceedings of the Fourth International World Wide Web Conference. O'Reilly, Boston, MA. 51-62
- [36] WorldWideWebSize 2007. <http://www.worldwideweb-size.com>, (accessed on May 2, 2008).
- [37] Wurman, R. 1989. *Information Anxiety: What to Do When Information Doesn't Tell You What You Need to Know*. Bantam Doubleday Dell Publishing Group. New York, NY.