

Personal information management tools

Joseph Schächli
joseph.schaeppi@unifr.ch

ABSTRACT

Vannevar Bush posited the Memex vision in 1945. A Memex is defined as a device where everything could be stored and retrieved. Forget-me-not, Stuff I've seen, Phlat, Haystack, Mylifebits are five different ways to implement this vision. They are different, but they share some good practices described in the personal information management literatures.

General Terms

PIM, tools

Keywords

Phlat, Stuff I've seen, MyLifeBits, Haystack, Forget-me-not

1. INTRODUCTION

Developing a tool for personal information management is a great challenge. Such a tool should have the ability to adapt itself to a lot of data and should provide views, searching features and filtering features which should be efficiently usable by everyone. The section 2 of this text will describe five different tools: Phlat, Stuff I've seen MyLifeBits, Haystack, Forget-me-not . The section 3 will present a list of works which are related to the tools presented in the previous section. Afterwards, the main differences and resemblances will be exposed in the chapter 3. Finally a short conclusion will present a summary of our discussion and some personal appreciations.

2. TOOLS PRESENTATION

2.1 Phlat

Phlat [?] is a project developed by Microsoft Research. It is a system that optimizes search for personal information with an intuitive interface that merges search and browsing through a variety of associative and contextual cues. Phlat combines keyword and property-value search in a seamless and intuitive manner, allowing users to find information

based on whatever they may remember, wherever that information may be stored. In addition, Phlat provides a facility for tagging content with user-created metadata to insure that users can return to their content.

Phlat uses the Windows Desktop Search indexing and search engine and provide access to its content. Phlat allow users to add hierarchical user created metadata , or tags to displayed content. Tags are directly associated with the files.

2.1.1 Design principles

Phlat is based on seven design principles:

1. Unify text entry and filtering. People are very comfortable searching with a few words of text and some filters. The proof is the frequent use of search engine like Google for our web search. Filter in Phlat is considered as a way to do a query.
2. Current search criteria must be visible and salient at all times.
3. Provide rapid query iteration. Result must appear quickly and the change must be obvious.
4. Allow iteration based on recognition. The user must have the possibility to refine his search with the information that is shown in the list of results.
5. Allow for abstraction across property values. For example, it's better to have only one filter for pictures than one filter per picture's extension (jpg, gif, etc ...).
6. Tag UI must support both tagging and filtering.
7. Integrate with common file system/mail operation. The user should be able to do common operation like copy, paste within the Phlat interface.

2.1.2 User interface

The user interface is composed of 3 main areas. The Query Area is in the upper left corner. It contains the query controls and a query box for entering and displaying queries. Below this area is the Filter Area which is made up of six buttons: Saved Queries, Date, Tags, Path, People and Type. Each button has a corresponding pane. In the right part of the user interface is the Results Area where search results are displayed in a columnar list view.

2.1.3 Evaluation

Phlat was evaluated by 225 people during 8 month within Microsoft organization. They observed that queries are generally short (1.6 words) and that approximately 25 % of all free-text queries include people's names. Filters are also often used. The people filter and the type filter are very often used. An interesting finding show that Phlat is used by many users as a tool for searching their email archives. The facility to do queries and that users didn't need to worry about where an item was stored were the two main points of satisfaction.

This evaluation has shown several real problems and some desires of users. The most common desire was to have better preview features. One of the main problem was the update of the index. Because Phlat runs on top of a search engine which is independent from the file and mail systems, a change into these systems takes some finite amount of time to propagate to Phlat.

2.2 Stuff I've seen

Stuff I've Seen (SIS)[?] is also a project developed by Microsoft Research. The main goal is to facilitate the re-use of each informations which have been previously seen. The system is composed of an index and of some searching features based on rich contextual cues such as thumbnails and previews. When a person sees an information, the system add it to the index. The system architecture of SIS is built on top of the Microsoft Search Indexing Architecture and consists of five main components: the Gatherer, the Filter, the Tokenizer, the Indexer and the Retriever. The Gatherer specifies the interface to different content sources in their native format. The Filter decodes file formats and emits a character stream for further processing. The Tokenizer breaks streams of character into words. The Indexer builds a standard inverted index structure. The Retriever contains the query language used to access stored information. Each of these five runs on the client machine.

2.2.1 User interface

The user interface allows a user to start a query, to refine it by filtering and sorting and to see the list of results. Two views were developed: the Top View and the Side View. In the Top View, filters are in the top of the layout and in the Side View, they are in the side of the layout. The Top View is conceived to be more flexible than the Side View. The advantage of the side View is that it is easier to understand for the user and that the position of the filters part allows to display more results. The disadvantage of this view is that it is not possible to filter by multiple types of items.

2.2.2 Evaluation

SIS was tested during a period of six week by 234 people coming from variety of background. This evaluation show that many characteristics of web queries are also present in SIS. Queries were generally short, averaging 1.59 words and 48 % of them involved a filter. 25 % of the queries involved people's name. 50% of the users begins the evaluation with the Top View and the other half begins with the Side View. 44 % of the people which begins with the Side View switch to the Top View. They were only 34 % in the opposite case. This effect is due to the richer search interface of the

Top View. Concerning the sorting, users have switched frequently to sorting by date. It suggests that Date is perhaps the most useful attribute for finding personal items. After installing SIS, most of the users have said that it was easier to find something and that they searched less frequently using native application.

2.3 MyLifeBits

MyLifeBits [?] is another project from Microsoft Research. It is a try to implement the Memex vision posited by Vannevar Bush in 1945 [?]. A Memex is a device in which books, records, and communications could be stored and consulted. MyLifeBits uses an extension of the Memex Vision which permits to handle audio and video, to perform database style queries, and multiple visualizations in the user interface. The main part of MyLifeBits is a database of ressources and links. Each ressource has the possibility to annotate another using a link. Several ressource properties including a short description are stored in the database.

2.3.1 Design principles

The design of MyLifeBits is based on four main principles:

1. A strict hierarchy must not constrain the user. An item could of course be classified into several category. For example, a picture of Fribourg belongs to the category pictures from Switzerland and to the category pictures from a town. The solution choosed by here is to allow each object to be assigned to zero or more sets of collections. This solution is only usable if we have excellent querying capabilities. A collection is defined here as a dynamic entity created via saved query. In other words, collections are implemented with annotations.
2. Many visualizations helps the user to understand what they are looking at.
3. Annotations are necessary to give a value to non-text media. An annotation may be also usefull to remind you of nuances of your past thoughts on the object. Annotations allow also the creation of stories. A story is for example a slide show or a photo album.
4. Finally authoring tools create two way links to media that they include in new media (transclusion).

2.3.2 User interface

Four views for the list of results have been implemented: detail, thumbnail, timeline and clustered time. A list of ressource including each property is display within the detail view. The thumbnail view show images of the ressources. Thumbnails are displayed on a linear time scale within the Timeline view. Within the clustered time, thumbnails are grouped by year, month, day etc ...). Figure ?? shows this last view. One of the goal of these four views is to provide information density to avoid extra clicks. Another goal is to minimized the action needed to have a sense of what something is. Therefore thumbnails are massively used. MyLifeBits offers many features to do annotation as easy as possible. It is also possible to create an audio annotation which will be converted afterwards into a text by the Microsoft Speech SDK.



Figure 1: The clustered-time view of query results used by MyLifeBits.

2.3.3 Evaluation

The SIS team have unfortunately not put a critical evaluation of their work in his paper [?]. Gordon Bell, one of the SIS team member has digitized almost everything possible from his entire life, but no evaluation's data is available from his experience.

2.4 Haystack

The main idea beyond Haystack[?] is that every person has his own way to work with information. Therefore Haystack gives to the user lots of liberty on how he want to record, view, annotate and manipulate information. Haystack stores arbitrary objects of interest by reference. Relationships between them and arbitrary properties are also stored. In Haystack Ressource Description Framework (RDF) is used to represent all the information. The model used by RDF is a graph in which the nodes are the information items to be managed and the edges are the relations we would like to record. Advantages of RDF are that it can be used without any schemata and that it has structural similarity with the World Wide Web. To import data in Haystack, a collection of extractors are applied to the traditionally formatted data.

2.4.1 User interface

Because the objects stored are not predefined, the user interface must be flexible. Haystack use \$view prescriptionT with a recursive approach to build the user interface. View prescription are data in Haystack and are customizable by the user. Concretely a view prescription is a collection of RDF statements describing the divisions of the display region, the graphical widgets used and what should be shown and where they should be. A critical point with this approach is to determine which view description should be applied. Haystack determine the view description using the type of object which will be displayed and the size of the area in which it will be shown and afterwards the view description with the best match is selected into the database.

The Haystack's view is remarkably impoverish. It offers only rectangular hierarchical decomposition, but it offers the advantage to show lots of different type of information in the same user interface. For example, a reporter could have in the same view all the stuff he needs for his next articles. It could be a list of pictures, a list of interesting contacts, some related texts and a list of interviews.

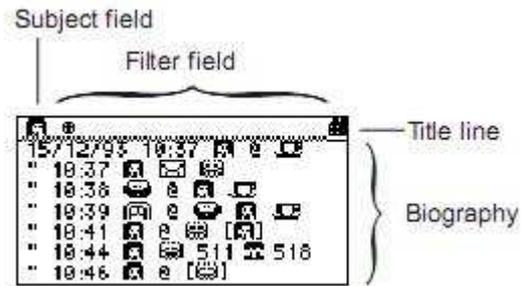


Figure 2: The user interface of Forget-me-not.

2.4.2 Evaluation

A limitation of Haystack is the user interface ambiguity. Sometimes, it is difficult to determine which object user is addressing with a given click. The reason is that every object in the screen is considered to be alive. For example, if someone clicks on the mail sender, should the mail object or the sender object be concerned.

2.5 Forget-me-not

Forget-me-not[?] is a project from the Rank Xerox Research Centre of Cambridge. It was developed in 1994 when mobile computing was in an early stage. It could be defined as a memory aid which helps to resolve everyday memory problems such as finding the name of the document that I have received during the last meeting. This tool is based on two fundamental concepts. The first concept is the episodic memory which is the process by which it is possible to remember an event starting from their context. The last concept is intimate computing. When we wear a computer, it makes sense that this computer behaviour is closed to our personal needs.

Forget-me-not resides in a small portable device called a ParcTab. Some informations about the user's activities are collected by this device and afterwards added into a personal biography. Each entities present in my personal biography have a private biography. For example, if the user takes part in a meeting , the meeting's participants, the start time, the location and the distributed documents will be recorded.

2.5.1 User interface

The user interface of Forget-me-not is very basic. It consists of a top line which is called the title line and the area below the title line which is the biography. Each entity is represented by a distinct icon. To filter his biography, the user will put the corresponding icons in the title line. This interface is shown on the figure ??.

3. RELATED WORK

Other tools for PIM have been developed. A first example is Lifestreams [?] which is a personal store that abandons conventional hierarchy in favor of time-based visualizations. Thomas Erickson developed a new personal notebook application [?]. Some task-centered management tools like UMEA [?] were especially create to captures the task context of content.

4. COMPARISON

The goal of this section is not to do an exhaustive comparison of these tools, but to find the main differences or resemblances between them. Forget-me-not is the only tool from the presented tools which was developed for a specific type of device (PDA). It is also the only one which has an ubiquitous approach. None of the others tools record automatically where is a meeting, who is at the meeting etc ... Haystack has the only user interface which could be dynamically customized for any type of information. Stuff I've seen and Phlat were created by Microsoft and built on another piece of software developed within this organization.

5. CONCLUSIONS

With the presentation of these five tools we have seen that it doesn't exist only one approach for the conception of a personal information tool. Nevertheless, each of this tool have in common with the other good practices. A good practice is for example to allow a filtering by people or by time. A problem which has not been solved yet is the privacy. These tools could record our entire life and have powerful searching and viewing features, but what happens if someone else access this data.

6. REFERENCES

- [1] V. Bush. As we may think. *interactions*, 3(2):35–46, 1996.
- [2] E. Cutrell, D. Robbins, S. Dumais, and R. Sarin. Fast, flexible filtering with phlat. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 261–270. ACM New York, NY, USA, 2006.
- [3] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 72–79. ACM New York, NY, USA, 2003.
- [4] T. Erickson. The design and long-term use of a personal electronic notebook: a reflective analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, pages 11–18. ACM New York, NY, USA, 1996.
- [5] S. Fertig, E. Freeman, and D. Gelernter. Lifestreams: an alternative to the desktop metaphor. In *Conference on Human Factors in Computing Systems*, pages 410–411. ACM New York, NY, USA, 1996.
- [6] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. MyLifeBits: fulfilling the Memex vision. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 235–238. ACM New York, NY, USA, 2002.
- [7] V. Kaptelinin. UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 353–360. ACM New York, NY, USA, 2003.
- [8] D. Karger, K. Bakshi, D. Huynh, D. Quan, and V. Sinha. Haystack: A general-purpose information management tool for end users of semistructured data. In *Proc. of the CIDR Conf*, pages 13–26, 2005.
- [9] M. Lamming and M. Flynn. Forget-me-not: Intimate computing in support of human memory. In