Calendar for Everything: Browse and Search for Personal Archive on Calendar

Satoshi Nakamura
Kyoto University
Yoshida Honmachi, Sakyo, Kyoto Japan
nakamura@dl.kuis.kyoto-u.ac.jp

ABSTRACT

The amount of personal content is increasing significantly. Consequently, it has become difficult for people to manage this content in their personal archives. As a result, almost all personal content has become dead storage. We have developed browsing and searching techniques based on a calendar interface with a personal archive and developed a system that recommends personal content. The combination of these techniques and system allows content to be retrieved from a user's personal archive.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentations (e.g., HCI)**]: User Interfaces – *Interaction styles (e.g., commands, menus, forms, direct manipulation), prototyping, user-centered design*

General Terms

Design, Human Factors.

Keywords

Calendar interface, recommendation, life log, personal archive.

1. INTRODUCTION

The number of digital files that people store has been increasing. This increase in stored files is because people are creating files as a result of engaging more and more in the following activities:

- Sending and receiving e-mail by desktop computer and mobile phone
- Taking photographs using digital cameras or mobile phone cameras
- Using portable music players
- Recording TV programs using HDD recorders
- Creating text and slides
- Using the Internet

A collection of such files containing personal content is known as a personal archive. It is difficult for a user to manage, browse and search for target content within a personal archive because a good environment for managing distributed personal contents does not at present exist. As a result, users usually never access their personal contents in their personal archive. Therefore, many personal archives become dead storage.

For example, the author stores 100,000 photographs, 50,000 e-mails and 3,000 documents in his computer, 1,000 songs in his portable music player, and 500 e-mails in his mobile phone. It is not uncommon for some users to store 0.1–1.0 million files. Initially, users manage files using a directory file system with the name of an event, group, and date. However, it is not easy for the user to classify many files in this way because there is much duplication. Most users eventually give up managing personal contents in this way. This means that users sometimes have difficulty searching target content.

Desktop search services, such as Google desktop search [1] and Windows desktop search [2], enable the user to search target content that is stored in the computer. However, the user cannot search for content that does not contain keywords. For example, it is difficult for the user to search image or video content because such content does not include much text information. In addition, the user cannot search the content if he/she does not know which query will direct the search engine to the correct target content.

However, users sometimes can remember something that is related to the target content. For example, a user may not remember the date a particular photograph was taken with an old friend; however, the user may remember something else he or she did on the same day. Such *neighboring* information can be used to find target content.

To solve the above problems, we have developed a calendar based user interface and recommendation system. Using our system, the user can browse in a calendar view all of a user's personal content stored in various media. In addition, the user can find target content from neighboring personal content by using a *neighboring* search function.

Our system can additionally recommend other personal content that is related to content that a user is browsing or editing. For example, when a user reads an e-mail received one year ago, the system recommends photographs that were taken one year ago or other e-mails that are related to the content of the email being read.

In this paper, we first explain the concept of our work and then give details of the calendar system and recommendation system. After that, we discuss the usefulness of our system. Finally, we provide a conclusion and go over future work.

2. CONCEPT

The goal of our work is to make effective use of a user's personal archive to support his or her daily life. We have developed two methods for effectively utilizing personal archives:

- *Active browsing/searching* using a calendar interface
- *Passive browsing* using a recommendation system based on the context of a user's active searching/browsing

Active search/browsing means that the user inputs an explicit query in order to find target content. If the user can easily create a suitable query, active searching/browsing is useful. In addition, the user can search for/browse target content using the date or neighboring content. However, the user cannot make effective use of his or her personal archive while doing something else such as browsing Web pages or writing e-mails because active browsing/searching requires interaction between the user and the personal archive browsing/searching system.

With *passive browsing*, the user does not input a query. The system uses context as the query. The system automatically recommends cross-media personal content that is related to the context of whatever the user is currently viewing, such as a Web page or e-mail.

Our system manages cross-media personal content according to the content's creation date and enables the user to easily check, browse and search everything via a calendar interface.

3. CALENDAR FOR EVERYTHING

Our system first collects all of a user's personal content, such as memos, schedules, diaries, e-mails, bookmarks, photographs, screenshots, scanned images, and movies. The content is taken from different sources, such as local disks, mobile phones, digital cameras and the Internet, and then stored in a daily-fractionated directory.

Our system also collects Weblogs and schedules by using the API of Web services. We only used our system with a Twitter, which is a micro-blog, and the Google Calendar, which is a popular scheduler. We did not classify any personal content depending on such criteria as its group, event and media type because of the difficulty of such a classification.

Our system automatically extracts meta-data from the personal content, such as a creation date, EXIF (Exchangeable Image File Format), information contained in a JPEG image, and comments on PNG images and movies. It then generates an index of all personal content in a directory based on dates.

Then, our system displays the personal content on a calendar interface (see Figure 1). Our system automatically changes the displayed contents at a preset interval. Our system has three display modes: daily mode (Figure 2), monthly mode (see Figure 1) and yearly mode (see Figure 3). The user can change the mode and the display period by simply clicking buttons. This enables the user to easily check personal content for a given target period.

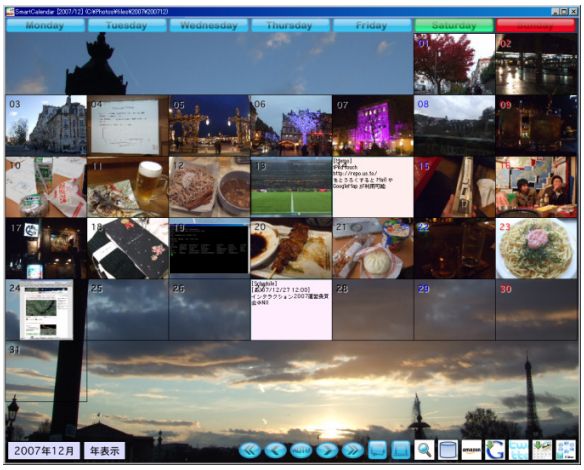

Figure 1. System shows text memos, text schedules, photographs, screenshots, and movies, etc. for each date in monthly mode of calendar interface.

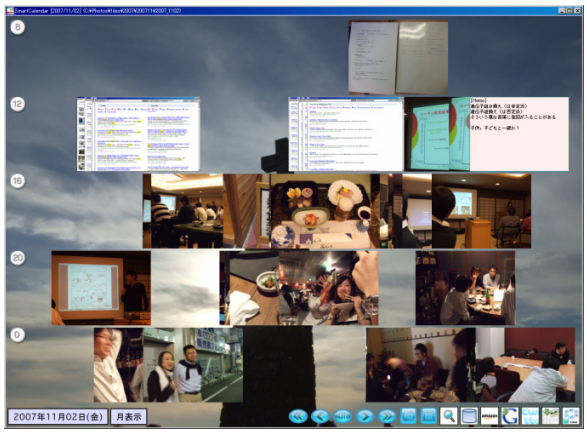

Figure 2. System shows personal contents in the daily mode.

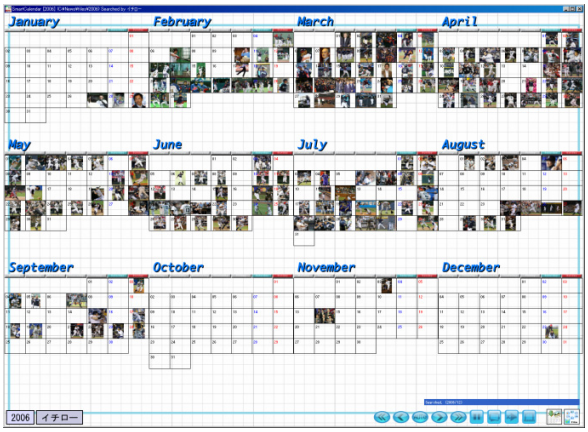

Figure 3. System shows personal content in each date in the yearly mode.

Our system enables the user to tag content by attaching text annotations, such as “lunch,” “slide,” and “funny”. In addition, our system enables the user to evaluate content with tags such as good/bad, like/dislike, and public/private. This means that the annotation is much easier for the user to perform. The user will be able to find target content easily with such annotations because our system propagates the user’s annotation to neighboring content. Our system puts such annotations on the target content. Our system enables the user to annotate personal contents by using a specific user interface (see Figure 4).

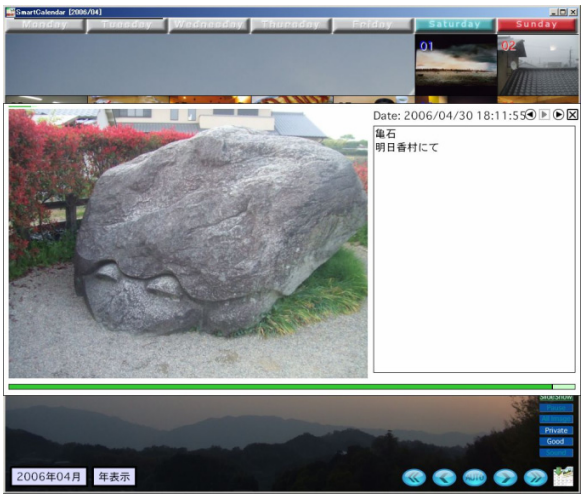

Figure 4. User can annotate target content by simply inputting text in this mode.

Our system has two types of search mode: a keyword based search and neighboring search of personal content. The keyword based search only displays content that satisfies the conditions of the query. For example, in Figure 5, the user inputs “meal” as a query and the system displays content related to breakfast, lunch and dinner for a given month.

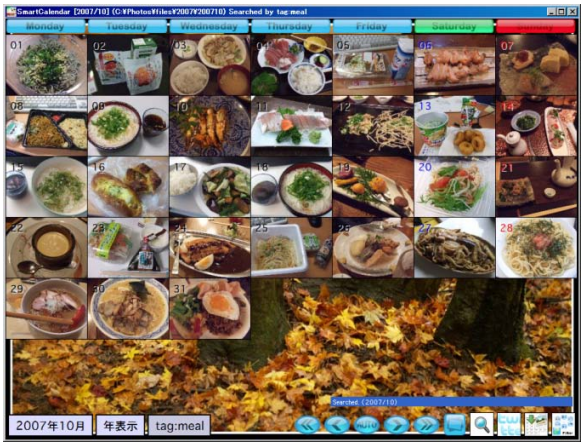

Figure 5. Search results for “breakfast or lunch or dinner” on the calendar interface. The user can find images of his or her family by checking what they ate.

In Figure 6, the user inputs “tag:face” as a query and the system displays the images of faces for the given month. The number of digital cameras that automatically annotate information for face detection has significantly increased. In this example, we only use such annotation. We can also annotate the image using not only the taking mode, white balance, and zooming mode but also the condition of the flash, ISO mode, etc. for annotation. We think it is real waste not to use such information.

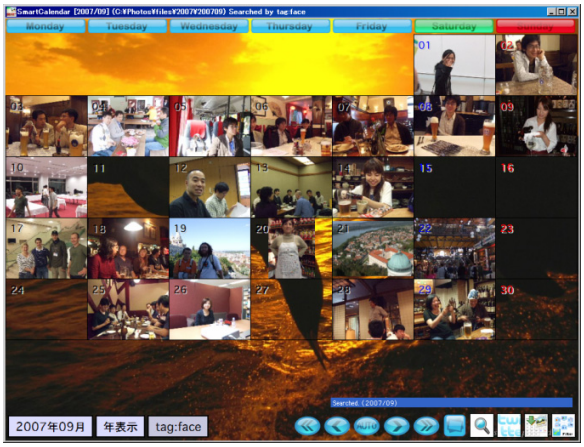

Figure 6. Search results of “face detected” on the calendar interface.

In Figure 7, the user inputs “time:10–13” as a query and the system displays personal content that was created, received/sent, taken, captured or scheduled between 10 a.m. and 1 p.m. The user can check his or her schedule, events or lunch appointments easily using this query.

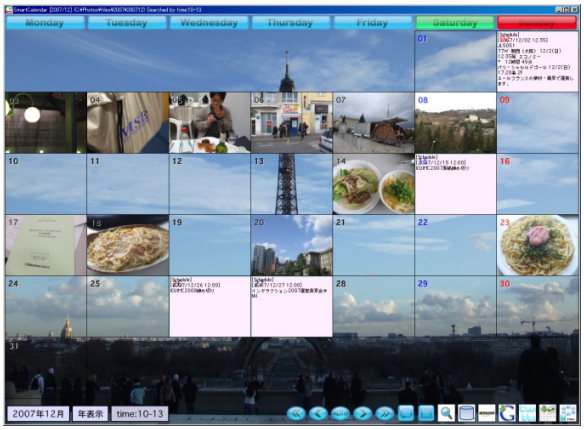

Figure 7. Search results of “time:10–13” on the calendar interface.

The neighboring search of personal content produces a display of content that is related to content the users is already browsing/searching. We assume that the user uses this search method in combination with a keyword based search. For example, in Figure 8, the user wants to find a photograph that contains the

user and his or her friends. However, the user cannot remember when the user met with these friends. When the user inputs the keyword for the event name, the system shows schedules that are related to the event name. After that, the user runs a neighboring search of schedules. Then, the system displays neighboring content in the result window. Finally, the user finds the target photograph from the window.

Figure 8. Combination of keyword search and neighboring search.

We have developed a *Calendar for Everything*. We implemented our system on Microsoft Windows Vista, Microsoft Visual Studio .NET 2005.

4. RECOMMENDATION SYSTEM

Our system detects the user's context from the opened window. For example, when the user browses the Web using a Web browser, our system obtains the title, URL, last-updated date, and the complete content from the browser application. Then, our system detects the topic terms. These are obtained by comparing the term-frequency and focused date. After that, our system automatically searches the calendar system using the obtained topic terms and dates and displays the related contents on the side-window of the desktop, as in Figure 9.

Figure 9. Our system recommends related personal content after detecting context.

When the user writes an e-mail to a person, our system obtains the name of the receiver and the e-mail's complete contents in order to obtain topic terms. Then, our system automatically checks the date of previous communications with the receiver and searches for photographs that contain the receiver's face and searches for related e-mails. Then, our system displays such related personal contents in the side-window of the user's desktop.

The recommendation system consists of a core module and four sub modules. The core module manages all of the processes and modules (see Figure 10).

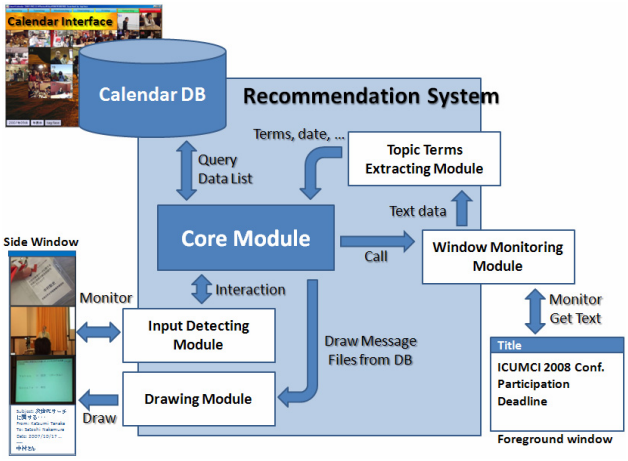

Figure 10. Design of recommendation system

The core module periodically sends a request to the window-monitoring module. In addition, the core module generates queries in order to obtain related content from the calendar database. The core module also sends a drawing message and content that are obtained from the calendar database to the drawing module. Additionally, this module regenerates the query based on the user's interaction and obtains related content from the calendar database.

The window-monitoring module is the module that obtains the title and the whole of the text of a focused window in order to detect context. This module checks the focused window after receiving a request from the core module. Then, this module obtains the title and the whole of the content from the focused window. After that, this module sends any obtained text to the topic-terms-extracting module.

At first, the topic-extracting module extracts frequent terms from the received text data using a Japanese morphological analyzer with a stop word list. Then, this module inserts frequent terms and personal names into the list of extracted terms. In addition, this module analyzes received text data by pattern matching in order to obtain date information. After that, the module sends extracted information to the core module.

The drawing module simply displays the received contents, such as text, image and movies, in a side window and recommends the contents to the user. In our system, this module displays part of a text, a thumbnail image or part of a movie.

The input-detecting module monitors user interaction. The side window allows the user to check complete content by left double clicking or to check neighboring content by right double clicking.

We implemented this recommendation system on Microsoft Windows Vista, Microsoft Visual Studio.NET 2005 and Mecab, which is a Japanese morphological system.

5. DISCUSSION

We developed our system and released it via our Web site[1]. Our system was also written about in some magazines. Our system has been downloaded more than 10,000 times, and we have received feedback from many users. We found that people could discover, remember and reuse content that was deemed to be dead storage. We think that our system is popular with users.

One user manages one million images (over 1 terabyte) using our system. This user has continued to use our system for a period of two years. The author also manages over 0.1 million files using our system and has continued to use our system over two years. Therefore, we can say that our system has scalability.

However, with the system as it is, the passive search is not so useful because of its lack of accuracy when extracting topic terms, delays in retrieving, and problems with overloading. To solve the problem of extracting topic terms, we are planning on improving the extraction algorithm. The delay in retrieving and problems with overloading are attributable to managing index information. We can easily solve these problems by loading all the index information into memory.

Our system does not work well if our system cannot collect the neighboring text information such as memos, diaries, schedules, and e-mails. Here, we found that our system is useful as a news viewer. In the Figure 11, the user input “Ichiro” as a query. The user then could check the the news related to Ichiro Suzuki easily by using our system. We are thinking of using such news as neighboring contents.

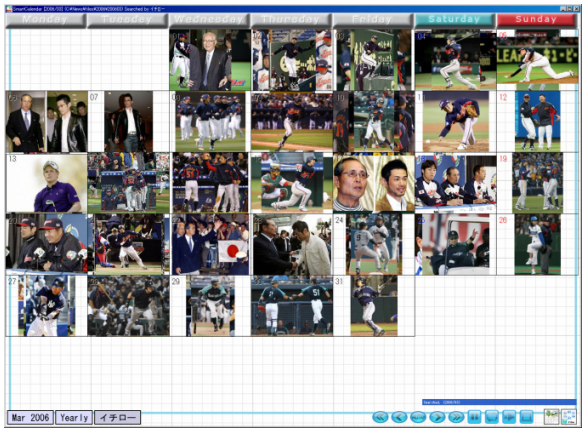

Figure 11. Examples of newsreaders

[1] http://calendar2.org/

6. CONCLUSION

We have developed a calendar based user interface system for use with personal content. Our system enables the user to manage and find personal content easily and recommends certain stored personal content that is deemed relevant to the context of the user’s search..

In future work, we are planning to do experimental tests in order to check the usefulness of our system. We also are planning to introduce a location-based neighboring search method and semantic neighboring search method. In addition, we are planning on improving our system to encourage the user to make annotations.

7. ACKNOWLEDGMENTS

This work was supported in part by MEXT Grant-in-Aid for Scientific Research on Priority Areas: "Cyber Infrastructure for the Information-explosion Era", "Contents Fusion and Seamless Search for Information Explosion" (Project Leader: Katsumi Tanaka, A01-00-02, Grant#: 18049041), and by "Design and Development of Advanced IT Research Platform for Information" (Project Leader: Jun Adachi, Y00-01, Grant#: 18049073), and by Grant-in-Aid for Young Scientists (B) "Content Manipulation and Browsing by Reversible Display" (Leader: Satoshi Nakamura, Grant#: 18700129), and by "Informatics Education and Research Center for Knowledge-Circulating Society" (Project Leader: Katsumi Tanaka, MEXT Global COE Program, Kyoto University).

8. REFERENCES

[1] Google Inc., Google Desktop Search, http://desktop.google.com/ja/

[2] Microsoft Corporation, Windows Desktop Search, http://www.microsoft.com/ windows/desktopsearch/

[3] Apple Inc., Spotlight, http://www.apple.com/macosx/tiger/spotlight.html

[4] Kaoru Misaki, Tools for Thinking about Myself and My Lifetime (Information Management Hacking), Journal of Information Processing Society of Japan, Vol.46, No.12 (20051215) pp. 1382–1386 (in Japanese).

[5] Takahiro Kawashige, Hiroaki Ohshima, Satoshi Oyama, Keishi Tajima and Katsumi Tanaka, Complementary Information for Email Messages using Context, 2006-DBS-140(2)-(73), Vol.2006, No.78(20060713) pp. 329–336 (in Japanese).