AUGMENTING USERS' TASK PERFORMANCE THROUGH WORKSPACE NARRATIVE EXPLORATION

A Dissertation

by

YOUNG JOO PARK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2009

Major Subject: Computer Science

AUGMENTING USERS' TASK PERFORMANCE THROUGH WORKSPACE NARRATIVE EXPLORATION

A Dissertation

by

YOUNG JOO PARK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Richard K. Furuta
Committee Members,	Frank M. Shipman, III
	John J. Leggett
	Steven M. Smith
Head of Department,	Valerie E. Taylor

May 2009

Major Subject: Computer Science

ABSTRACT

Augmenting Users' Task Performance through

Workspace Narrative Exploration. (May 2009)

Young Joo Park, B.S., Yonsei University;

M.C.S., Texas A&M University

Chair of Advisory Committee: Dr. Richard K. Furuta

In a fast-paced office setting, information workers inevitably experience expected and unexpected interruptions daily. As the volume and the diversity of information and application types grow, the impact of frequent interruptions on their task performance gets more severe. To manage the negative effects of interruptions on work performance, workers often engage in task management activities to ensure they are better prepared to resume suspended task less stressfully. However, managing tasks causes additional cognitive burden and a time cost to users who already are experiencing the tight attention and time economies.

This dissertation presents an approach to augmenting users' task performance by allowing them to manage and retrieve desired work contexts with ease. The Context Browser, the implementation of the proposed approach, is designed to help the users to explore narratives of their workspace manner and restore their previous work contexts. The goals of implementing the Context Browser are to 1) unload the users' burden of taking care of their task-related or task status information promptly and thus help them focus solely on executing a given task, 2) allow them to browse their previous

workspace intuitively, and 3) enhance continuity of their tasks by supporting them to retrieve desired work context more quickly and easily.

In order to validate the proposed approach, a user study comparing task performances of the group with the Context Browser to the one without the Context Browser was conducted. The study produced both quantitative and qualitative results. The study confirmed that with the Context Browser subjects expressed better quantitative numbers than the ones without. Subjects using the Context Browser were able to restore and retrieve their desired work setting and task-related information more quickly and correctly. Qualitative results showed that the subjects using the Context Browser found that various contextual cues and the interfaces responsible for providing the cues offered effective artifacts to help them recover both cognitive and work contexts, while the other subjects experienced a difficult time in restoring the desired contexts that were necessary to perform their assigned tasks. In addition, we re-invited 6 subjects from the group without the Context Browser 6 weeks after the study. We asked them to perform the same tasks as the ones they did 6 weeks before with the Context Browser. It showed that with the Context Browser they outperformed their previous performance even after a lengthy period.

DEDICATION

To My Father

ACKNOWLEDGEMENTS

First of all, I would like to thank my wife, Eun. Without her support and presence in my life, I could not have made this happen. And my two kids, Sean and Leslie, gave me the reason why I should keep trying no matter how hard it is and gave me the miraculous fatherhood which quickly became much harder and more special than my Ph.D. work.

Next, I thank my advisor, Richard Furuta, who helped me build a solid research project and take the right direction. I also want to thank my committee members, Frank Shipman, John Leggett, Steven Smith, for their time and feedback.

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

LIST OF TABLES

1 INTRODUCTION

In the realm of computer and digital resources, performing a task requires using a variety of information entities due to the growing diversity of applications and the Internet. People whose jobs are carried out using a computer usually perform multiple tasks a day. During performance of those tasks, many different information objects come and go over their computer desktop. In that sense, the desktop of a computer serves as the interface to numerous individual digital information resources existing in local hard disk drives and on the World-Wide Web. This parallels the use of information in a traditional information worker's work-setting, since while performing a task, a user must organize, search, use and create multiple information resources.

Tasks can vary from job-related tasks, such as developing an application or planning a new marketing strategy, to personal ones such as writing an email or shopping at online stores. In particular, a structure of a significant task among job-related tasks tends to be complex since one of the important characteristic of such a task is requiring significantly more documents (e.g., associated information resources), which possibly makes the task's structure more complex [Czerwinski et al., 2004] than one with less associated resources. If users leave the task unfinished due to either an expected or unexpected interruption, they often find difficulties in resuming the task because they encounter factors impeding continuity of the task. When an interruption occurs, a relationship between the task and a set of associated information resources can be easily broken because the task's environment, i.e., desktop status, starts being

---

This dissertation follows the style of *ACM Transactions on Information Systems*.

adjusted to the new task's requirements – users need to close some of the current windows and open new ones. With the changes to a users' working environment, their previous context fades away and a new one emerges, which means that they may start losing the reference point retaining the previous context.

Task execution often includes a user's activities that were not planned, but instead were adaptive ones to the current task context. Such activities vary from Web search practices with a variety of keywords to frequent document modification due to instant content changes. Users' activities are highly situational and context-based processes that are also subject to the constraints of the working environment [Barreau, 1995]. Suchman [1987] claimed that each action is closely associated with "local interactions contingent on the actor's particular circumstances." Hence, there are many information uses that are activated by immediate requirements and are not implementations of a ready-conceived plan, because these interactions occur at the time when the user needs them. These ad-hoc information uses easily slip away from a user's memory and can hardly be reinstated at a later time; reconstructing those information uses can be a very tedious job.

2 PROBLEM

For information workers, performing tasks typically coincides with frequent interruptions and resumptions, which leads to the need for an efficient mitigating interface for minimizing discontinuity and task management. To develop such an interface, we have paid attention to helping users maintain continuity of, and manage, their tasks with ease.

2.1 Discontinuity: Lack of Support for Maintaining Task's Continuity

As a computing environment evolves and the diversity of information and applications grows, the internal structure of a task becomes more complicated. In particular, a significant task tends to be complex and to last a longer period of time, which makes the task vulnerable to an interruption [Czerwinski et al., 2004]. Interruptions that users cannot avoid have been known to impact users' task performance negatively; we call this phenomenon discontinuity. Discontinuity in task resumption stems from several factors such as complexity of a task's structure, duration of an interruption, users' imperfect recall, multiple primary computing devices and so forth. If the previous task had not been completed prior to switching to the new one, then it might cause a foreseeable difficulty when they try to resume the task later – as the previous task's context will need to be reconstructed first. If there is a way for users to resume the task in the very contextual environment that was used earlier, then they can manage the task's resumption less stressfully, i.e., without tracking down the task history [Smith and Vela, 2001]. Unfortunately, the current computing environment does not support this—does not provide any organized information regarding how the tasks have been carried out and how the tasks have evolved. When an interruption lasts for a long period or previous working context is fairly complex to be restored quickly due to the maturity of the task, it may cause severe discontinuity to restoring previously used settings and starting the task. Further, when users carry out a task placing heavy workload on them, an interruption to the task can cause more severe annoyance [Iqbal and Bailey, 2005]. As such, diverse factors stand in the middle of users, tasks, interruptions and resumptions, and severity of discontinuity between an interruption and resumption can be measured by 1) the time to refresh the users mind and retrieve desired resources and 2) the length of a path directing to a proper context where the users can efficiently resume the task. Therefore, to help them maintain a task's continuity, we need to provide users with artifacts or clues that may shorten the time lags and ease mental workloads to restore users' cognitive and work contexts.

When resuming a task, the users often ask the question, "where was I?" They try to find valuable artifacts that help them recall what they were doing before the interruption. The best guess thus far or the most powerful clue for the aforementioned question can be found in the last moment of a computer desktop, i.e., workspace, when s/he left the desk or suspended the task. According to an initial survey that we conducted prior to an actual experiment, 86% of interviewees answered that they often left the computer turned on while they were away from their desk. We could easily imagine that many users wanted to keep the last desktop status alive until they need to come back to their computer since the last status was expected to play roles of task reminder and context restorer before the resumption of the suspended task. Since the current desktop environment lacks support for buffering the damage from interruptions, leaving a computer powered on is a cognitively light, casual and efficient scheme at the moment, in order to resume the task later with less effort. Unfortunately users have multiple tasks in their hand, which means that the last status can hardly provide clues for the other tasks.

2.2 Hardship of Task Management

Bannon et al. [1983] argued more than two decades ago that users often switched around their multiple tasks. Further, we are not talking only about just a simple task such as writing an email or reading news from the Internet, but also about the users' primary tasks that are lasting longer and are suspended more frequently [Czerwinski et al., 2004]. Then, let's go back to the question of "where was I?" and the argument of keeping the last moment of a computer desktop. Since multiple tasks are in users' hands (which is a usual circumstance for information workers), they commonly switch between those tasks during the day. With that, to be able access the last moments of all previously suspended tasks, there should be an interface retaining the corresponding last moments of all previously suspended tasks. In this vein, there have been efforts [Bannon et al, 1983; Henderson and Card, 1986; Tashman, 2006] to support users in maintaining multiple workspaces or in managing task-related information resources, which we normally call task management. In a bid to support efficient task switching activities, users naturally engage with the act of managing task with or without the aforementioned task management tools. Paradoxically, it is left up to the users to manage task context that will make sense to them later. The users, however, already have enormous workloads in performing given tasks each of which is fighting for the users' attention and time. In a fast-paced office environment, users can hardly afford devoting time to managing tasks since they are already experiencing a tight attention economy and a high volume of information. Many users who perform multiple tasks daily are vulnerable to interruptions, task switches and task resumptions. Under such an environment, it doubles their cognitive load because they have to deal with both task management and the task itself without an appropriate any mitigating interface to those difficulties. Barreau [1995] points out that work processes can hardly be fit into document-oriented categories such as subject and title. Having them focus just on what they are doing and what they will do after the current task would be a much more desirable choice, instead of placing an additional burden on the already overloaded individuals. We thus need to give a peace of mind to users by saying “Just do your work. We will take care of the rest.” Apparently, task management is just one more thing to do for users [Hudson et al., 2002].

3 RELATED WORK

3.1 Management of Tasks and Interruptions

Information workers usually have multiple tasks on their hands, which forces them to interleave those tasks. In this regard, Gonzalez and Mark [2004] and Bannon et al. [1983] argued that design of information technology should understand the fact that people are constantly switching between different working spheres. Henderson and Card [1987]'s *Rooms* was the initiative of modern virtual desktop managers. However, users need to build their own scheme to switch around tasks and handle interruptions causing them to switch to a next task. Interruptions plague overall task performance and user concentration since they make it difficult for users to maintain an on-going task's continuity. O'Conaill and Frohlich [1995] conducted a study and reported that many interruptions result in discontinuing the interrupted work beyond the actual duration of the interruption. Mark et al. claimed that information workers generally work in an average of 10 different working spheres and they stay about 10 minutes in a working sphere before switching to another one due to high fragmentation [2004]. They also reported later that information workers often experience work fragmentation due to interruptions and their study discovered that 77.2% of interrupted work was resumed on the same day [2005].

Czerwinski et al. [2004] reported that they found characteristics of a key project that distinguishes it from a user's various ones. Those are 1) it is lengthier in duration, 2) it requires more documents, 3) it experiences more frequent interruptions, and 4) it is revisited more often by a user after an interlude. Further, they found that users have hard time restoring their focus on task after their attention is altered. The timing of interruptions particularly affects the users' emotional state [Bailey et al., 2001], which may affect whole task performance[Adamczyk and Bailey, 2004]. If we cannot avoid being interrupted, then to minimize the inappropriate effect from interruption, Iqbal and Bailey [2005] suggested that interrupting at the most proper moments consistently caused less resumption lag and annoyance for particularly work-aligned tasks. Iqbal and Horbitz [2007] argued a major problem faced by users is to restore the context of the suspended tasks, mainly when there were multiple applications in the suspended work context. They also recommended an easy access to the suspended application contexts should be a key feature for the design of recovery tools. There have been many studies [Bardram et al., 2006; Dragunov et al., 2005; Dumais et al., 2003; Kaptelinin, 2003; Oliver et al., 2006; Robertson et al., 2004] on users' task management. Commonly they tried to provide an interface to support easier retrieval of their workspace and their desirable information. However, requiring users to spend their time on managing tasks places a burden on the overloaded individuals and a system should buffer the demand of management [Hudson et al., 2002].

3.2 Information Organization

Information workers inevitably organize information to help themselves complete given tasks. There are two major trends in information organization which are *pile* and *file* [Kidd, 1994; Lansdale, 1988; Malone, 1983; Whittaker and Hirschberg, 2001]. Malone [1983] also mentioned that the cognitive difficulty of categorizing information heavily affects how people organize information – people are often reluctant to file information away because they are not able to decide which category at the time. He thus suggested that a computer-based system may handle the difficulty by embracing both characteristic of piles and files. According to Dumais and Landauer [1983], piles are compensating strategy for the problem of files, i.e., classification. Lansdale [1990] claimed that the use of piles help people identify target information by visual recognition, spatial property and a time without performing the process of classification. However, he also pointed out that the scanning process which is one of primary retrieval processes becomes less efficient as piles of information accumulate. Mander, Salomon and Wong [1992] investigated how people handle the flow of information in their workspace. They discovered that users often group documents spatially and process information by forming physical piles of paper, rather than immediately categorizing it into desirable folders.

3.3 Effective Contextual Cue

In this section, we briefly introduce studies to identify artifacts which help users to remember and more generally to restore tasks suspended earlier. Several studies [Graham et al, 2002; Kelly and Davis, 2003; Krishnan, 2005; Monty, 1986; Plaisant et al., 1996; Rekimoto, 1999] argued that a time or a temporal attribute is considered an effective cue that helps users recall what they were doing at the time. In addition, they visualized their information environment to augment their task performance since a time-based interface enables the users to discover a large amount of important information, not to mention related data. Ringel et al. [2003] tried to extend a timeline view by adding public and private landmark properties and they discovered that a group with the properties complete search tasks more quickly than the group without them.

To improve users' recall, several studies utilize visual and spatial properties. A study reported by Blanc-Brude and Scapin [2007] shows that visual elements (e.g., graphics, pictures, color, etc.) are one of the most effective attributes to determine the correctness of recall. SenseCam project [Sellen et al., 2007] shows that images can be an effective memory cue and particularly passively captured images possibly cause users to remember more events than they do with their own actively captured images. Data Mountain [Robertson et al., 1998] found that allowing users to use iconic representation of documents using 2D spatial layout takes advantage of spatial memory. Its extended study [Czerwinski et al. 1999] uses visual highlighting cues to group web pages which are semantically related and showed faster web page retrieval time. WindowScape [Tashman, 2006] exploits users' spatial and visual memories by use of a thumbnail layout by which the users can access multiple windows in multiple tasks.

Fass et al. [2002] argued that human memory for certain information is affected by the context where the information was presented earlier. They also found that improving human memory by use of the concept of context requires the users to put significant effort into creating the context, making the context distinct and relating the context meaningfully to what they are doing, which are not the tasks they are willing to perform. Infocockpit [Tan et al., 2001] uses location and place as primary cues to augment human memory. In the same vein, cognitive scientists found that environmental context effects on memory are actually reliable [Smith and Vela, 2001]. Typically elements of environmental context may include the time, the place in which the information was acquired earlier, smell, sound and so forth [Smith and Glenberg, 1978]. For instance, if people are placed in a same physical environment where they had a meeting previously, then they have a better chance to recall what they talked about than ones are in a different environment. Kidd [1994] indicated that retaining the physical context on the users' desk can be used as an effective tool to reinstate a complex set of threads without difficulty and delay.

4 APPROACH

An interruption is the main culprit that impedes users' overall task performance since it raises issues such as task management, task resumption, and recovery of both cognitive and task contexts. Those issues, however, are intertwined; each of them can affect the other ones. As mentioned in Section. 3, there have been several studies regarding the aforementioned issues. However, since we believed there was still a room for an improvement to enhance users' information work practice, we initiated this research to aid users in such a way in which they can perform tasks with less time and mental costs.

4.1 Worry-free Mode: Supporting Flexible Range of Task Management

Task Management is essentially about classification, filing and retrieval. Working with the desktop metaphor, even today, requires putting its conventions and rules into users' memory instead of interacting intuitively. Malone [1983] identified two types of strategies for handling paper-based information: *filing and piling*. Filing represents a neat desktop and piling represents a messy one. As he concluded, the act of classification, i.e., filing, coincided with a serious cognitive load, which we can easily imagine from how much effort users made to make the desk neat or messy. Further, Landsdale [1988] pointed out that organizing information "involves psychological processes." It is a quite natural reaction for users not to spend a great deal of time to filing information because it has no tangible immediate advantage and because they want to begin the next piece of work. Therefore, it does not surprise that users often try to defer organizing information, regardless of task-related or personal information, as long as possible. However, piling can unload several difficulties stemming from frequent classification by use of a scanning process – a typical search scheme for piles. Surprisingly, Whittaker and Hirschberg [2001] found that “filers amassed more information, and accessed it less frequently than pilers”, which gets us rethinking about the usability issues regarding filing and piling methods in an information worker’s context. As a result, we considered piling the compensating strategy to organize task-related information more casually.

To resolve this fundamental issue, we found the need to support a delayed classification rather than asking users to categorize task-related information resources promptly. In an effort to ensure that users can file their information away at their convenience, we need to add an automated piling module archiving users’ use of information automatically. With deferred or at-any-time filing being available, users can simply focus on performing a task rather than managing what they are doing to prepare what they will do later. In a bid to support that, we embrace both filing and piling methods into a task management process. Piling has been known to require less mental effort than filing does. However, filing is a superior information organization method leading to better organized and therefore easily accessible archives. We want to acquire all the benefits from the both information organization methods.

4.2 Retrieval Tag and Bookmark

In the previous section, we discussed the use of two different information organizing methods, filing and piling, to support a flexible range of managing task-related information. To avoid negative effects stemming from the very act of filing, an automated piling method was selected to express an idea of “we will keep it for you, so then you can manage it whenever you want to." We then directed our attention toward a way of embedding user-defined, not system-generated, structures into piles. According to the experiment conducted by Lansdale et al. [1987], the degree of semantic link significantly affected users' recall performance. In particular, users assigning retrieval tags to documents showed much stronger recall rate for those ones, i.e., user-defined tags, as opposed to having them assigned automatically by the system, i.e., system-generated tags. As such, we believe that allowing users to tag their information serves as a semantic filing scheme and a direct and efficient retrieval cue as well. In doing so, users can incrementally and conveniently convert piles into files that are becoming semantically categorized.

When users make a significant progress to the task that was tagged previously, they may want to keep the current state under the same tag, which is achievable by letting each tag point to multiple states, i.e., bookmarks. This example clearly stresses the fluidity of users' context and opens the door to bookmarks to manage the fluidity. A bookmark is a tool to hold any significant state and to let users access the state with ease. Users regard tags and bookmarks as surrogates for their task and workspace.

4.3 Providing Effective Contextual Cues

To avoid "where was I" problem, Kidd [1994] indicates that spatial layout of materials on a user's desk provides very effective and immediate contextual cues to reinstate a complex set of threads without difficulty and delay. A study reported by Blanc-Brude and Scapin [2007] showed that visual elements (e.g., graphics, pictures, color, etc.) are one of the most effective attributes to determine the correctness of recall.

A user's desktop status is an ultimate visual abstract and contextual cue that expresses a user's use of information. By archiving the visible changes of window arrangements of a desktop, our system can facilitate retaining a visual narrative of users' working environment, in which they performs various tasks. In doing so, a pile of the narratives contains various contexts used for different tasks, and functions as an information organization allowing a user to easily browse a timeline view of, bookmark, and retrieve any of their previous working contexts, even after a lengthy interruption to a task.

4.4 Making Archives Searchable

The goal of this research about helping users find what they were doing, thinking and planning to find a way to go forward. In this vein, all of the aforementioned ideas were trying to provide interfaces with which users can discover the contexts that they want to be in less stressfully. However, there are cases where users should locate something that they never tagged and bookmarked and they don't know whether or when they ever used it. It then does not leave many options, except a search interface asking for user-created keywords to locate it. A search module would shore up the users who are trying to find something that was not considered important or reusable and whose value was not determined at the time. In a sense, this most common and important feature in an information service arena is a key wheel of the vehicle fundamentally driving the message that "There is no worry. We will keep it for you, let me know when you want to retrieve or use it."

5 USERS' DESKTOP, WORKSPACE AND CONTEXT

While performing a task using a computer, a user is typically situated in a context in which s/he usually encodes 1) environmental variables referring to the features of a physical place where the task was carried out and episodic incidents that happened while performing the task, 2) materials on a user's desk as well as ones on his/her computer's desktop and 3) mind settings such as what a user was planning to do or thinking of at the time. Psychology researchers have shown that people could have a better chance to recall information when they were surrounded by the same environmental background in which they obtained information earlier [Smith and Vela, 2001]. For example, imagine that you had a meeting with a client at a restaurant a few days ago and that you want to remember a certain part of previous conversations you had with the client. If you go back to the restaurant and are surrounded by the same background (smell, sound and sights) at the time of the meeting, then it will improve your recall.

Next, consider the work context in which a user handles various information resources necessary for the task that has been interrupted by other events. Later, in order to resume the task from where s/he left off previously, the best way is to bring him/her back to the exact state when an interruption happened, i.e., the last moment. This exact state may consist of both the same physical contextual environment (desk, stuff on the desk, environmental features, and such) and the same digital contextual environment (a computer, visible information object on a desktop, files, running applications, and so on); see Figure 1. A physical contextual environment generally does not change much daily. The reasons behind this might be that it is hard to be reconstructed once it is disrupted or changed, or the users will engage in similar tasks for the time being. On the other hand, on a digital contextual environment, a user's activities over the environment leave traces within a user's computer and those traces can be used to reconstruct the prior state of the contextual environment. Further, the traces might also serve as a memory cue to help remind of other contents related to the task. Of course, if a user switches to a different computer and wants to resume a previously interrupted task, then the switching can be another factor that increases discontinuity. However, in the digital domain, it is possible to reconstruct the state in a different computer if the corresponding data is transferable (portable) from the original computer to a current one.

Therefore, this research is describing a user's context largely based on a user's digital contextual environment, i.e., a desktop, in which a task was carried out. As a result, a computer desktop plays a role of a contextual environment, and diverse activities over the desktop are identified as context-dependant information.

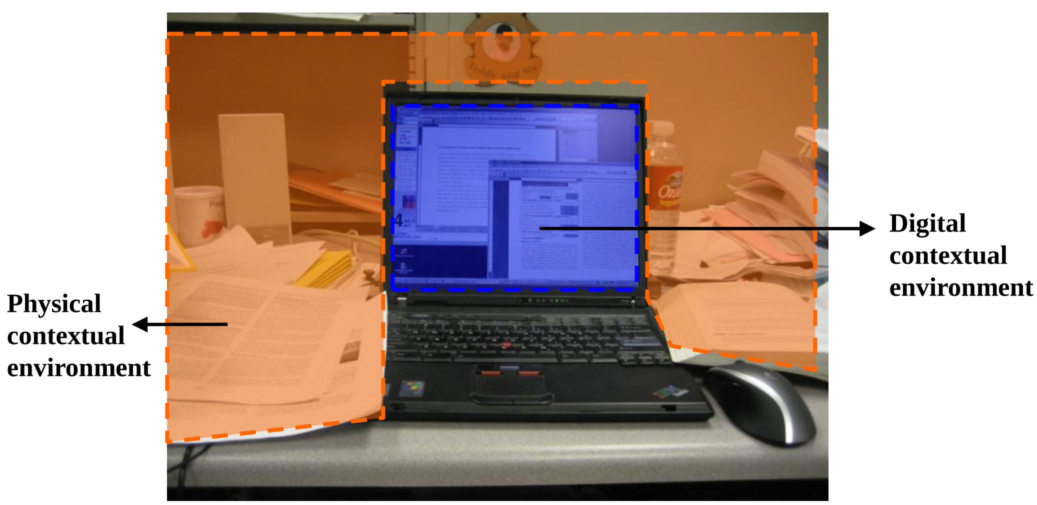


Figure 1. Typical work settings

5.1 What Is a Context?

A *context* in this research is defined to be a task-oriented concept that typically carries information about 1) what users' task (s) is (are) and 2) which information resources have been used to perform the task, i.e., context-dependant information. However, each context is not necessarily mapped to single task since people can execute multiple tasks in parallel in which they arrange multiple work settings on a desktop. Basically, a context informs users what was (is) happening on their workspace. As the task progresses, the user makes changes on information objects on a desktop, i.e., windows, to meet updated requirements of the task. Context-dependant information forming the users' work context contains particularly 1) various information resources visible on a desktop, 2) which applications (or windows) were used to read and change information resources, 3) which files were used and what were the changes if made, and 4) which URLs were visited to gather information from the Internet.

When switching to a different task from the current one, the users may start arranging different information objects on a desktop – a context switching may begin. As a desktop displays different information objects, the users lose the work context used for the previous task. However, we believe that contextual information, such as desktop status, contains very effective contextual cues, and making the users' work contexts subject to retrieval can augment the continuity of a task and ease the cognitive burden on them.

6 DESIGNING THE CONTEXT BROWSER

Context Browser is an implementation of the idea of "placing a user into an appropriate context in which the user can carry out a task with less cognitive and time costs." The context browser, as described in Figure 2, consists of two major tiers: 1) *Context Archiver*, collecting and saving a variety of events during task execution and 2) *Context Retriever*, retrieving and presenting contextual information to the users. In this section, we describe the architecture of the context browser and its components.

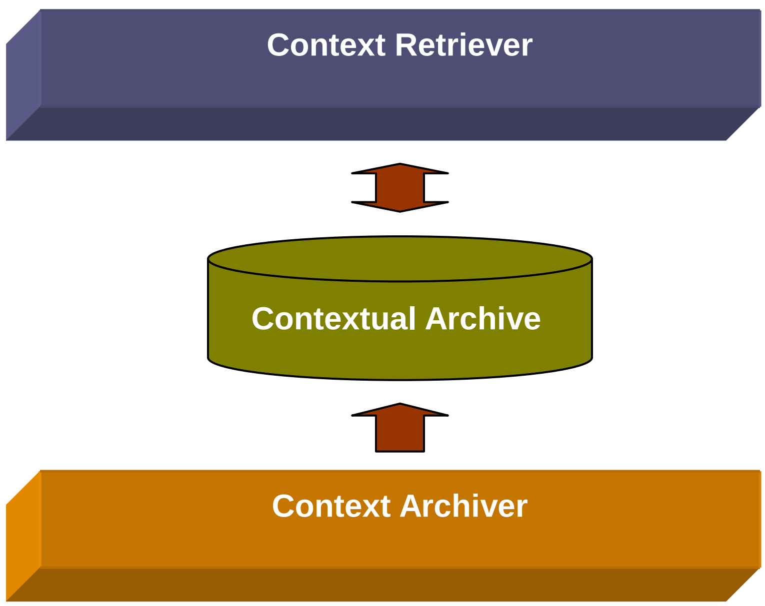


Figure 2. Two main tiers around contextual archive

6.1 Context Archiver

Context Archiver generates a contextual information archive containing a collection of metadata of users' activities. It gathers archival sources from various event monitors each of which watches and logs a designated type of event. The archive will be

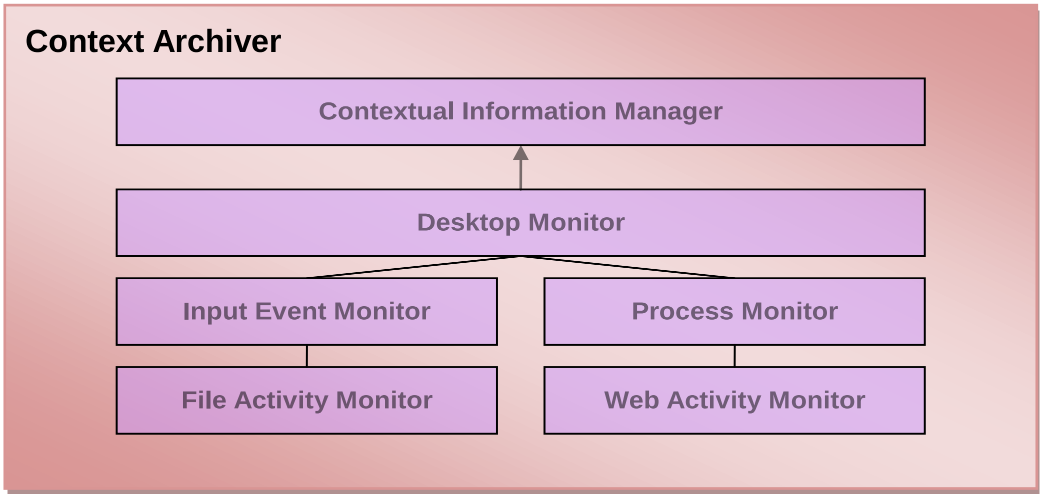


Figure 3. Internal components in the context archiver

accessed by the context retriever to help the users identify their work context at a later time. With use of the above monitoring modules, it continuously archives desktop status and users' activities on the desktop to eventually formulate a time-based data pile. There are five monitoring components that are *desktop monitor*, *process monitor, file activity monitor, input activity monitor* and *Web activity monitor* and one archiving module, *contextual information manager*, grouped into the context archiver together, (see Figure 3). In the following sections, we will describe how the aforementioned and other components in the context archiver work in concert and help the system generate the user's contextual information archive.

6.1.1 Desktop Monitor and Process Monitor

Desktop status naturally conveys a work context expressing the users' use of information and reminding of their thought and plan at the moment for a given task. The users eventually launch and use window objects, i.e., applications, that are needed for executing the tasks and each application holds its own a target information resource such as a file and a Web page. When they launch a window object, i.e., an application, the process monitor, which keeps an eye on a process call event, detects that a new process has been invoked and informs the desktop monitor. The desktop monitor then explores all of the currently opened windows on the desktop to see if any newly created window associated with the newly launched process has been added to the desktop. If it successfully confirms that a new window object has been actually created on the desktop, then the desktop monitor stores the window object's properties such as its main process name, its location, its size and its target information resource at the time. As seen in Figure 4, the desktop monitor collects and maintains metadata used to express desktop status.

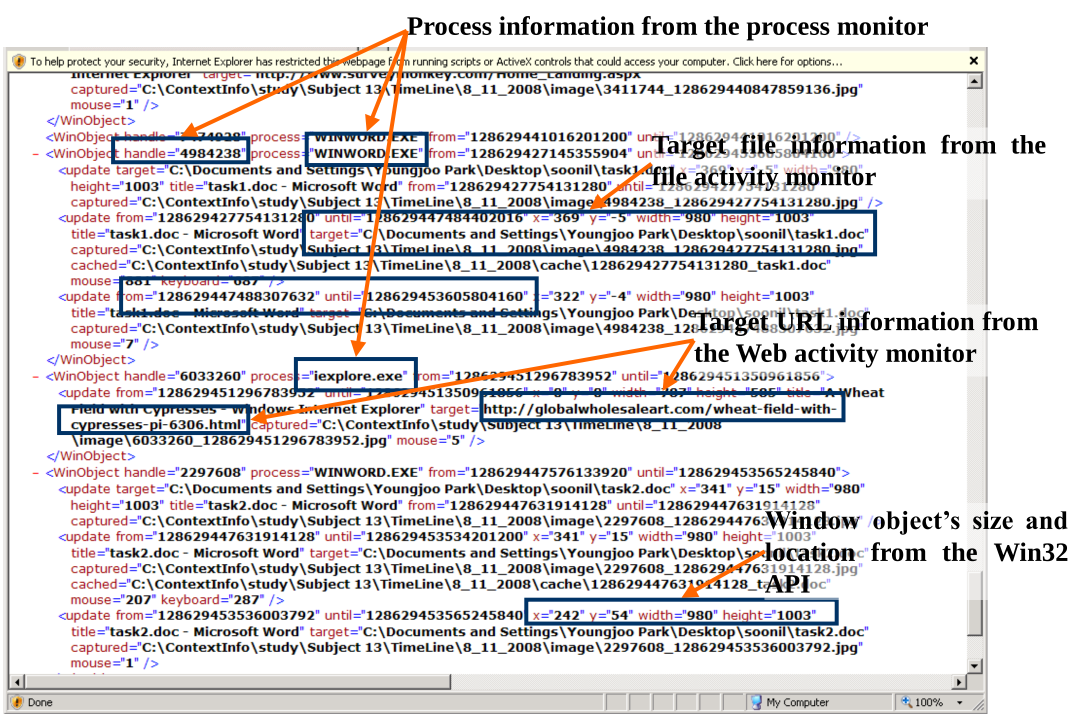


Figure 4. Metadata elements representing desktop status

6.1.2 File Activity Monitor

In developing the context browser, the most onerous part was to sort out and log file events. A file activity monitor utilizes the *FileSystemWatcher* class which is defined in the .NET framework and allows us to watch all file events fired due to the events of *created*, *deleted*, *renamed* and *changed*. But with the capacity to listen to all types of file events, we also had to deal with filtering out irrelevant ones, which occur so frequently. For instance, when a user launches the MS Word application, numerous files (most of which of course are application-specific reference files) fire *file_change* events due to

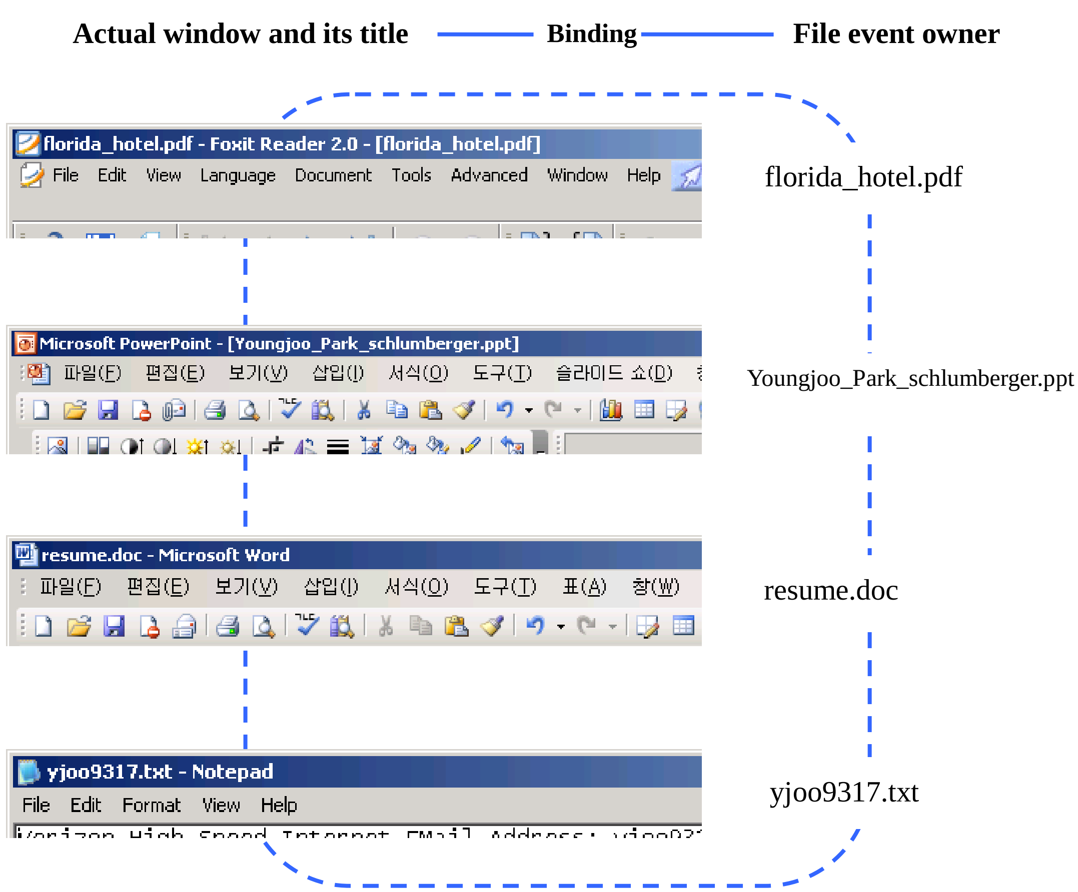


Figure 5. Binding an application to the target file

the system's attempt to access those files in the course of launching the application. As a result, the context browser starts receiving an enormous quantity of the events from no-need-to-look files that obviously are not relevant to the users' task. In an effort to block the non-related file events, the file activity monitor ignores the events fired from files in specific directories, such as "Program Files", "WINDOWS", "Temporary Internet Files" and so forth, because the contents of those directories are mostly system-dependant or application-dependant files, not user-created ones.

Even though it managed to stanch the flourishing stream of file events, there is still a problem of finding out which application has opened which file. For example, when a notepad application has opened a file "manual.txt", there is no gratuitous service API to let us know that "manual.txt" was just opened by the notepad application whose HWND (window handle ID) is 988574382. Virtually it is about fiddling with a black box since the Windows operating system's open programming interface, i.e., Win32API, does not allow us to access that part of information. To bind a newly reported file with a corresponding window object, we took a heuristic approach to resolve the aforesaid problem. As we all know, the title of any window always represents a name of the current working file or the identity of the content of the window. Windows' titles, which are easily obtainable by use of the Win32 API, are responsible for showing key features of their contents, mostly file names. As seen in Figure 5, when the context browser receives a message that a new file has been accessed, the file's name is compared to the titles of all windows on a desktop to identify whose title shows the highest relatedness. Then the file is pronounced that it belongs to that window.

A file object associated with one of the above events may run as either a background or a foreground task. When a file object associated with one of the window objects fires an event, it reports the event to the desktop monitor. Then the desktop monitor updates the status of the corresponding window and logs the change. There are also cases in which a file object has been accessed by a non-window process, i.e., a background process. For instance, when a file is being downloaded from the WWW, there is no window object related to the file. In such a case, it logs the event as a pure file event separate from desktop status.

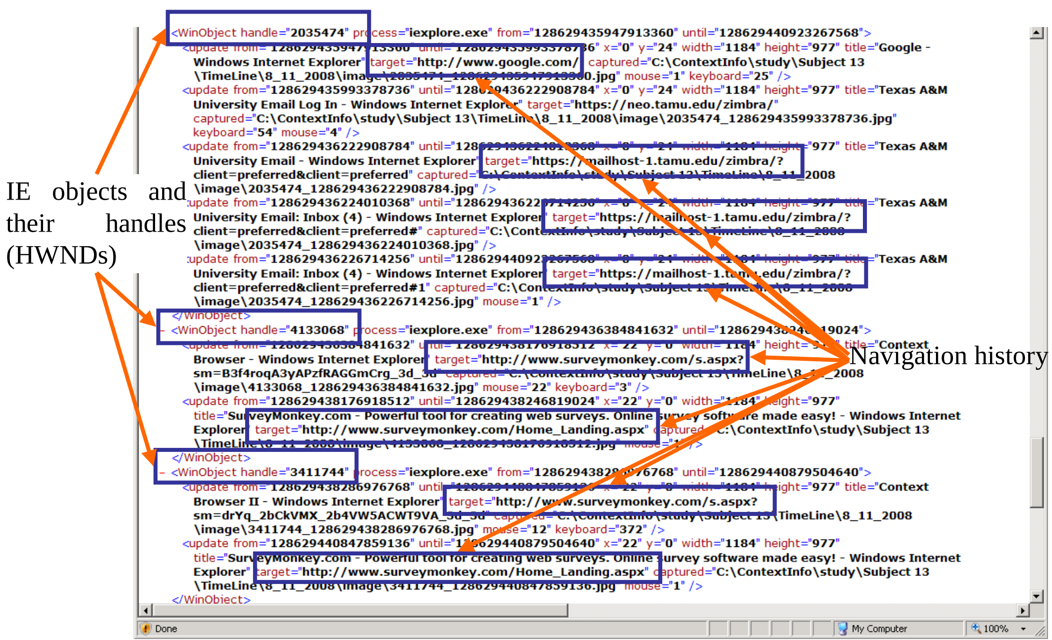


Figure 6. Navigation history of IE objects in the archive

6.1.3 Web Activity Monitor

The Web activity monitor listens to the events fired from any Internet Explorer windows via the COM interface – Extending to the other browsers, i.e., Firefox, Chrome and such, is achievable by implementing browser-specific extensions to help the context browser support various Web browsers. The current version collects a user's Web activities only from active IE objects on a desktop. In particular, when it fires the *Document_Complete* event which is issued when an IE completes rendering the Web page, the Web activity monitor updates metadata to mark that it has moved to a new page. As a user navigates Web pages, it records their URLs to maintain the user's navigational sequence of any IE object, see Figure 6. In addition to that, it captures rendered pages, which will be presented as either a thumbnail view or an actual size view on the Web activity viewer upon his/her request. It eventually maintains a navigation sequence not only of the individual IE object based on its window handle, i.e., HWND, but also overall the user's WWW activity.

6.1.4 Input Activity Monitor

This monitoring module observes a user's input activities fired by mouse clicks and keyboard strokes. When the context browser runs, it activates global mouse and keyboard hooking modules to enable the system to catch a user's input activities. It catches on which window object a user is clicking and typing and retains the frequencies of both mouse and keyboard inputs both globally (over a desktop) and locally (over an individual window), see Figure 7. These frequencies may carry meaningful values for the system to determine which information element, i.e., a set of a window object and its target resource, drew a user's attention most while performing a given task. Moreover, as users switch windows in temporal sequence during performance of a task, it maintains the window switching matrix that will aid to determine which windows are semantically related to each other. In a later section, we will further discuss how this matrix will be utilized in search activities.

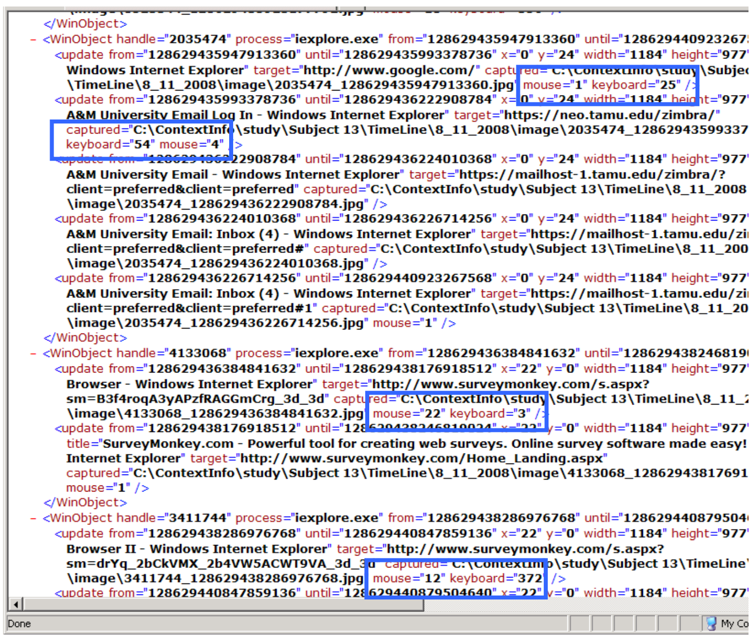


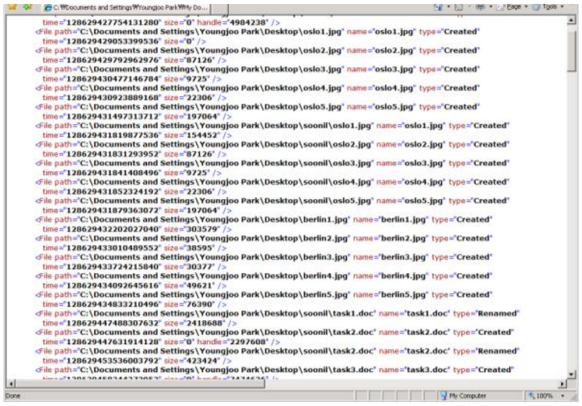
Figure 7. Mouse and keyboard input frequencies logs

6.1.5 Contextual Information Manager

This manager is a final outlet to an actual information archive. It receives various information packets from the aforementioned monitoring modules. As a way to create the archive, it generates XML files that contain various metadata elements regarding desktop status and the user's activities logged along with timestamps. There are three separate metadata files that are responsible for maintaining timeline data, tag/bookmark data and windows switching logs respectively, see Figure 8. To provide the narratives of a desktop in a time-based manner, it stores both desktop status and the user's activities that are essential to describe a desktop's narratives. The dataset necessary to represent desktop status at the time includes a set of running window objects' properties such as the following:

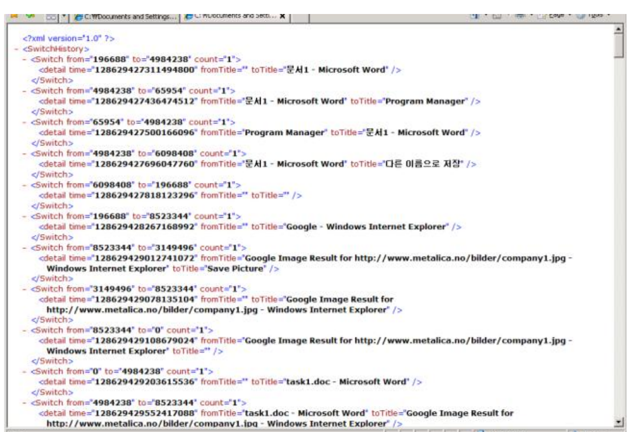
- Application's name
- When it was launched
- How long it had been running
- Its location and size
- Its target information unit (file or URL)
- Captured window image
- State change such as target file change or loading new URL
- Input frequencies of mouse and keyboard

Since a user carries out given tasks using necessary applications, a primary information entity that forms the desktop status is the window object. Hence, it stores which application was used and which target information unit was accessed by the application. It also records the information on how often the users interacted with their workspace by logging the frequencies of mouse and keyboard inputs. In addition to that, it separately logs file activity events to cover any files that were not associated with any window object on a desktop, for instance, downloading a file from the WWW.



a. Logging file activity

b. Logging windows on a desktop



c. Logging window switching history

Figure 8. Metadata files to render overall desktop status

6.2 Context Retriever

While the context archiver collects and stores information artifacts forming narratives of a desktop, the *Context Retriever* is responsible for bringing the context to users. It allows the users to explore, modify and add structural properties to the collection and to most importantly make the collection retrievable. The context retriever consists of four separate components eventually working in concert, see Figure 9. Hereafter, we describe those components and their functions.

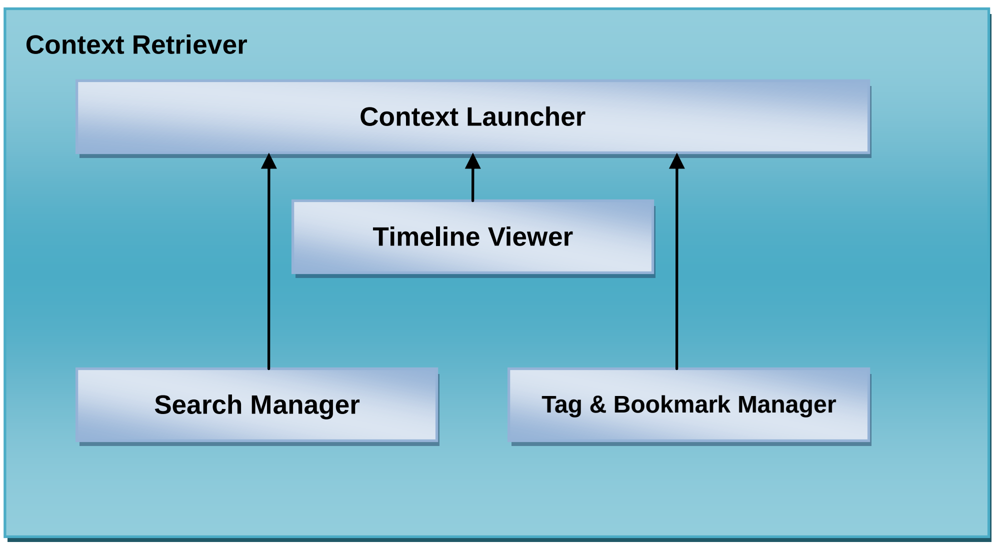


Figure 9. Internal components in the context retriever

6.2.1 Timeline Viewer

The *Timeline Viewer* is the main user interface that users use to explore narratives of a desktop to find their desired context, or any necessary information resources, as seen in Figure 10. It essentially employs and visualizes a pile metaphor to allow the users to scan, and to intuitively interact with the stack of previous desktop contexts in which they previously performed various tasks in either sequence or parallel.

The stream of various interaction logs mentioned in the earlier section is visualized and placed on a timeline. Users can further modify those artifacts seen on the interface to build a customized work setting in which they can continue any unfinished tasks. Detailed description about how a user can interact with this browsing interface and invoke necessary interfaces packaged into the context browser will be discussed in the following section.

6.2.2 Tag & Bookmark Manager

The tag and bookmark manager handles the requests from users to tag and bookmark desktop status while either performing a task (i.e., current status) or browsing a timeline (i.e., previous status). With this manager, users can store and manage their

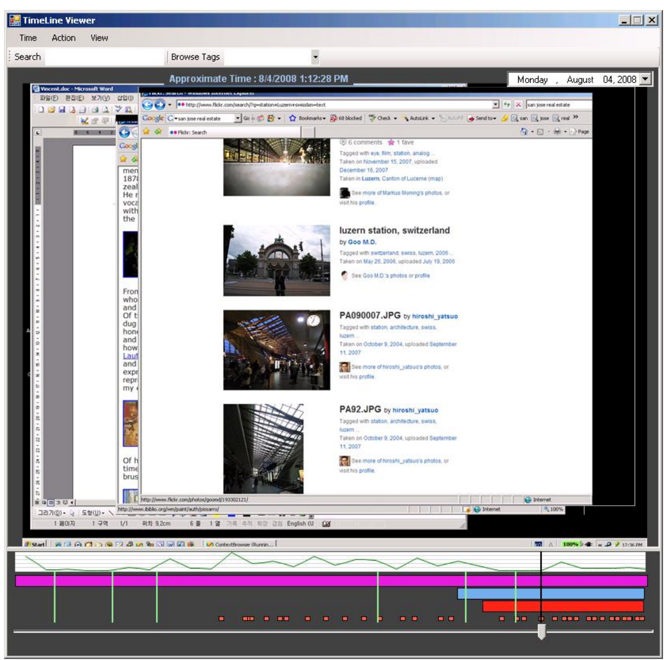


Figure 10. Timeline viewer

working contexts and retrieve them using the tags and bookmarks, (see Figure 11). From a task management perspective, tags represent thematic values telling literally what they are doing using their own words (not predefined or controlled words) and bookmarks can be used namely as markers indicating important states of given tasks (or tags) so that users may easily return to any of the marked states later. Putting it simply, a tag may represent a task and a bookmark may serve as a quick entrance to a certain state or progress of the task without a need to browse around the corresponding timeline. Each tag is allowed to have multiple bookmarks to acquire varying states of the tag. Therefore, when bookmarking a desktop status, users can associate the status with a tag that can be either a new tag or existing one. Basically, a tag and a bookmark are tools to shape a time-based archive in such a way that users can easily find a desired work context. In the following section, we will discuss more about how to use these features in the scene.

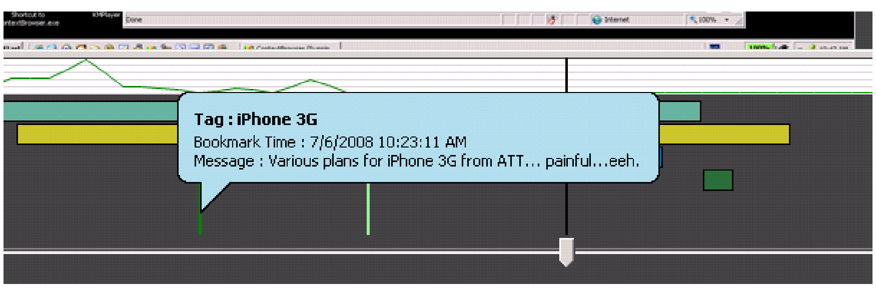


Figure 11. Tag and bookmark information on the timeline interface

6.2.3 Search Manager

In modern computing, users are inevitably exposed to an overloaded quantity of information and search is the most needed feature to retrieve something they thought important or something they missed previously. The fundamental information

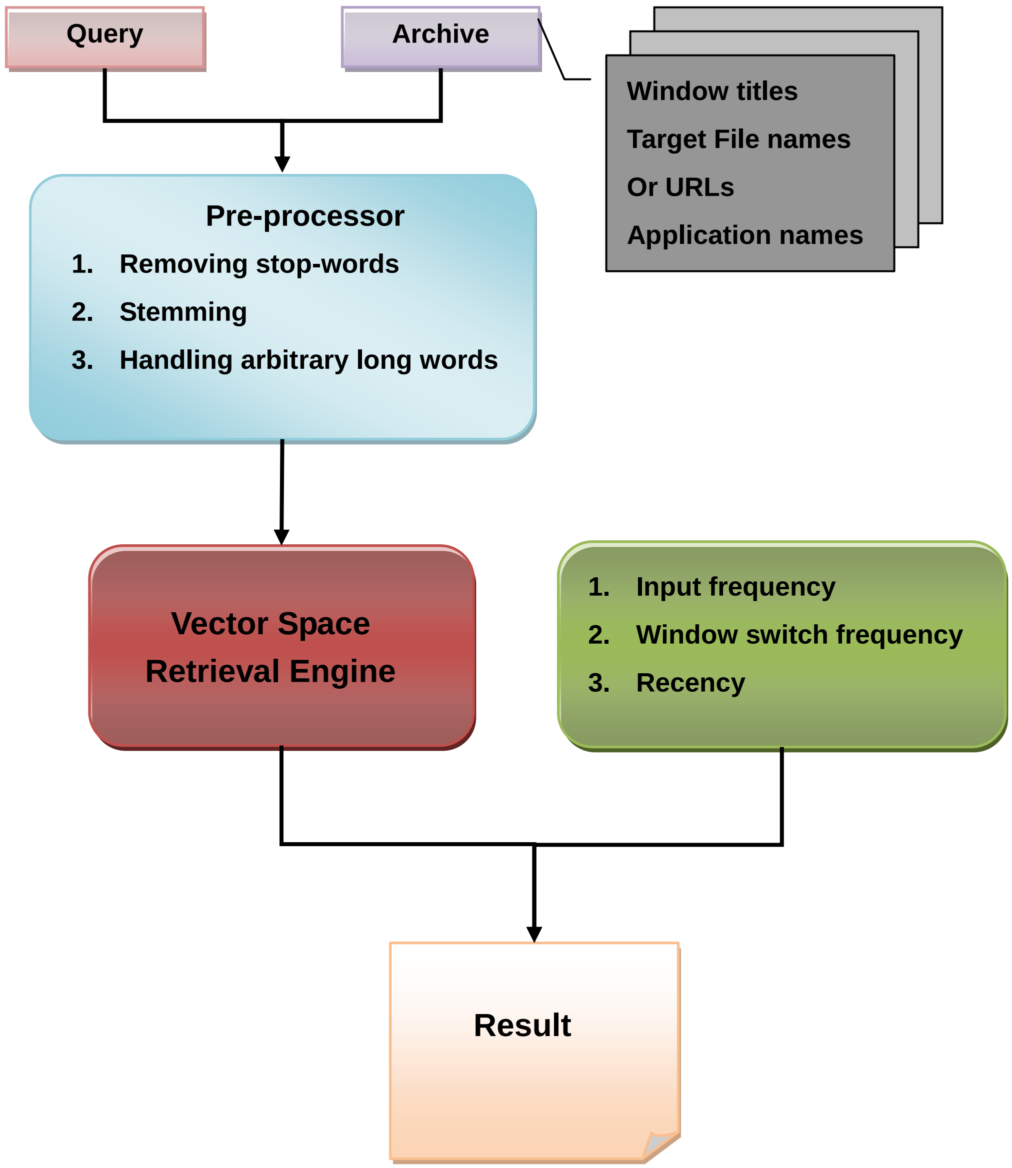


Figure 12. Architecture of a search engine

organization is a pile which is loosely structured and designed to spare users from having the burden of frequent information management. However, to handle a weakly structured information collection, a search module is a critical vehicle to lead users to the portions in the pile containing desired information.

Figure 12 shows the architecture of the search engine in the context browser. The contextual archive that will be compared against a given query is a collection of textual information: 1) a window title, which often reflects a target file or Web page's title, and 2) target file names or URLs. In our research context, source words are limited to those included in the aforementioned collection and are short in length. Due to the scarcity of data, it is very important to make sure that the system obtains every information-carrying word. To ensure that, when a query is entered, the query and the archive are fed into a pre-processor to achieve the following:

1. Eliminate stop-words: In addition to the typical stop-word categories, such as preposition, conjunction, articles and so forth, text in window titles often contain application-specific words that have no relationship with a window's content. For instance, "Microsoft Word" appears in every Microsoft Word application window's title. In this phase, we also eliminate those application-dependant words unless those appear in the given query.
2. Stem words: We need to have generalized terms to build meaningful term vectors for the terms in a query and those in corpus. It generalizes words in both in a query and the contextual archive by stemming them prior to converting them into term vectors. Stemming scheme has been implemented based on Porter's scheme [Porter, 1980].
3. Shorten long words: It is not rare for the text in a window's title or a file name to become a long string resulting from trying to compose a meaningful name with multiple words, such as "2008_winter_travel_seattle_vancouver.doc". Ideally, we want to partition such a long-worded name into smaller meaningful units, i.e.,

"2008", "winter", "travel", "seattle", "vancouver". A splitting module in the system splits long string into multiple smaller components. To achieve that, it detects any known word linking symbols, such as "–" and "_", and then splits the string composed with those symbols into multiple words.

After these pre-processing steps, we have neat versions of windows titles, file names, URLs and the Web pages' titles and these pre-processed words are ready to undergo the next step, a term vector analysis.

When a pre-processing has been complete, the query and the corpus are passed to a vector space retrieval engine, which has been implemented based on a vector space model [Salton et al., 1975]. The most basic representation of the document corpus consists of the raw frequencies of occurrence of terms in documents, i.e., *tf*. As generally known, this representation has the disadvantage that commonly occurring terms may unnecessarily make all documents look similar even though they are not characteristic of a particular document. Therefore, the majority of modern search engines apply the inverse document frequency measure, i.e., *tfidf*, which is $tf \bullet idf$ where *idf* is the inverse frequency of documents [Aizawa, 2000]. This measure adds a weight to the raw frequencies to compensate for the aforementioned disadvantage. In other words, it tries to scale down frequently occurring terms and scales up words that rarely occur. However, since a given query is compared against a set of short textual information, such as file names and window titles, all of which usually do not show high frequencies of any given terms, we did not utilize the *tfidf* measure as we would have in the context of a full-document model. After having cosine measures between the term vectors calculated, i.e.,

$\cos\theta = \frac{v1 \bullet v2}{\|v1\|\|v2\|}$, we additionally used a user's input frequency, which may represent their interest or at least a level of interactivity, and a recency as ranking factors to prioritize the items that have similar similarities. Therefore, when there are total *n* windows searched under a given query, the rank for window *i*, R_i, is defined as the following:

$R_i = s_i + \frac{u_i}{u} + 0.1 \times \frac{r_i}{r}$, where s_i is the similarity between two vectors, u_i is a user's input frequency on the searched window i and $u = \sum_{i=1}^{n} u_i$, and r_i is a timestamp of window *i* and r is the most recent timestamp among n searched windows.

Window switching frequencies maintained by the input activity monitor have been used to represent semantic relatedness between windows on a desktop, which is particularly meaningful in terms of searching highly related items to the one that users are currently viewing from the searched list. We will describe further detailed use of this search module in a SECTION 8.

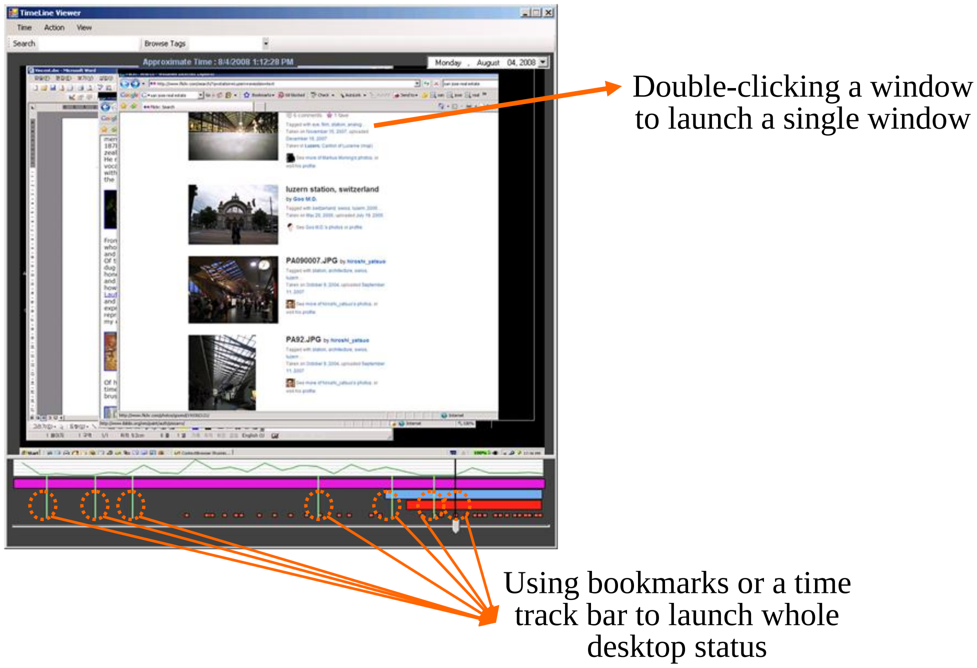


Figure 13. Examples of two different types of launching a context

6.2.4 Context Launcher

The context launcher is a simple window process launcher upon users' request to restore their desired context. The request comes from the timeline viewer, the tag and bookmark manager, and the search manager depending on which interface the users used to retrieve the context that they want to restore. The request packet could be containing either a single window object (a user double-clicked a single windows) or a set of windows (a user decided to launch the selected desktop status), see Figure 13. Upon the request, it creates a process and passes a file name or URL as an argument to the process.

7 NARRATIVES OF A DESKTOP: TIMELINE BROWSING

In this section, we describe a context browser's main user interface, a *timeline viewer*, and how users can interact with it. A context browser is designed essentially to help increase the continuity of a task by providing rich contextual information in a time-based manner. To achieve that, the system collects and visualizes narratives of a desktop which we define as the history of various interactions between users and their computer.

7.1 Stacking up and Browsing Desktop Narrative as a Time-based Pile

To keep the context of the use of a computer, Fass et al. [2002] stated "the user should put effort into creating the context" in order to improve the context's ability to be an effective memory aid, but they also noted that this could be a burdensome task. In an effort to unload such users' burdens of organizing information and let them access previous contexts on a desktop less stressfully, the context browser keeps track of every happening on a desktop without users' direct interventions and provides the users with a time-based browse interface. The timeline view interface, see Figure 14, is the primary interface for users to browse narratives of their workspace. As users move around a time track bar, it restores desktop status at the selected time. In general, users often revisit their primary tasks daily and even revisit several times within a same day [Mark et al., 2005]. In this regard, a time serves as a very effective contextual cue [Plaisant, 1996; Rekimoto, 1999] and there is an immediate benefit from the use of the automatically piled-up flow of desktop status since they can easily access a previous context used for the prior task by scanning the time-based pile. A reconstructed desktop in the timeline view interface supports typical desktop interactions with a mouse such as selecting and

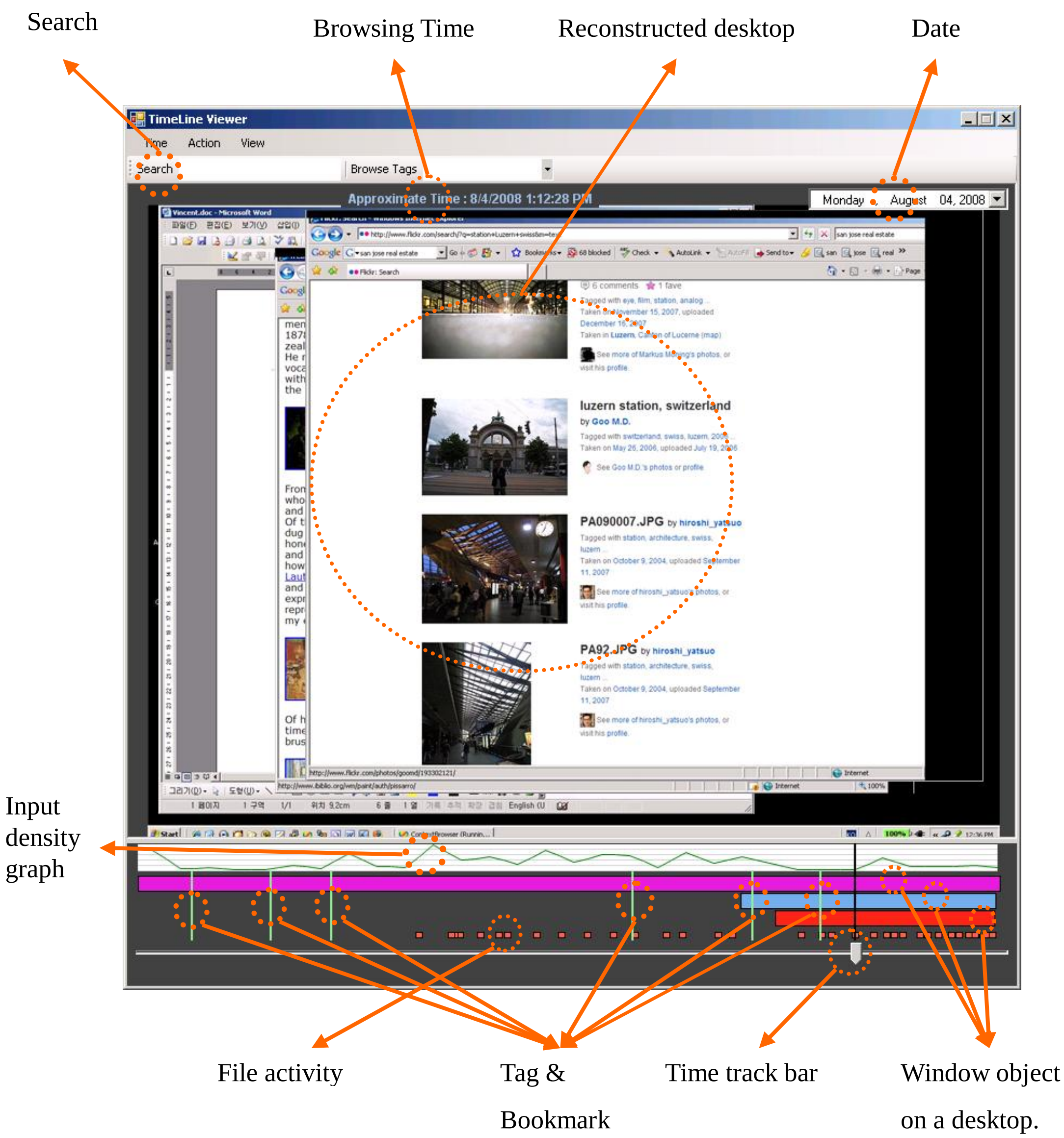


Figure 14. The timeline viewer interface

dragging around windows, and even double-clicking to launch a window. In addition to that, it supports launching the whole desktop status to retrieve a desired work context as opposed to launching an individual window, which means users can resume any previously interrupted task from the state when a prior interruption happened to the task.

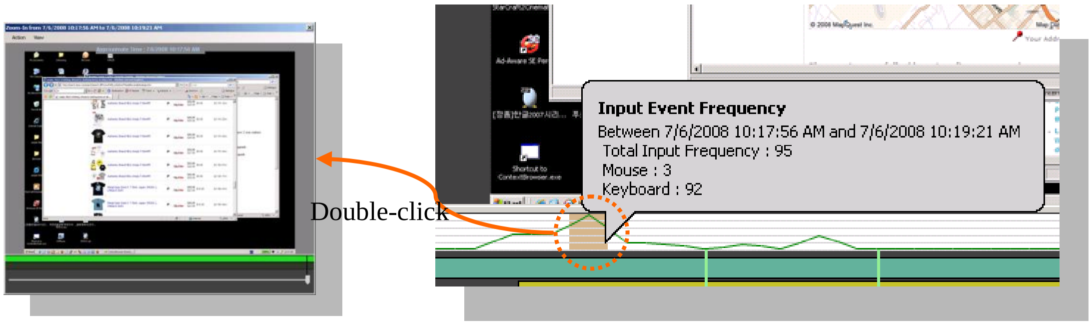


Figure 15. Input density graph (right) and zoom-in viewer (left)

There is a graph representation of mouse and keyboard input frequencies by a time, i.e., an input density graph, which is designed to help users recognize during which time frame how deeply they were involved with tasks. We assumed that a high level of interactivity might suggest that meaningful progresses could be made during such a circumstance. As shown Figure 15, when a user hovers the mouse over an input density graph, it shades the corresponding time frame area and exposes the frequencies of the mouse and the keyboard in parallel. If they double-click the area, it launches a zoom-in view interface that delivers the zoom-in view for the selected time frame. The timeline view visualizes narratives of a desktop during the given day. When an active duration of a desktop is, say, 10 hours, it eventually omits such windows that lasted too short to be shown since the number of window objects during the day can reach tens of windows, which eventually consumes unnecessarily much of real estate of the context browser interface. In such a case, it exposes only such windows that lasted for a relatively long period and users see only an abstract of the flow of desktop status on a given date. Yet by use of the zoom-in view interface, they can actually see every detail of desktop status

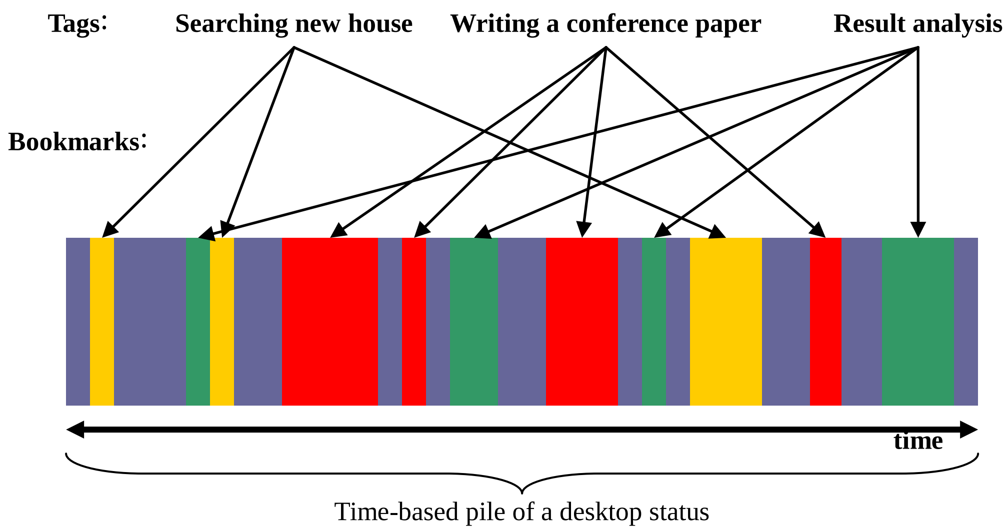
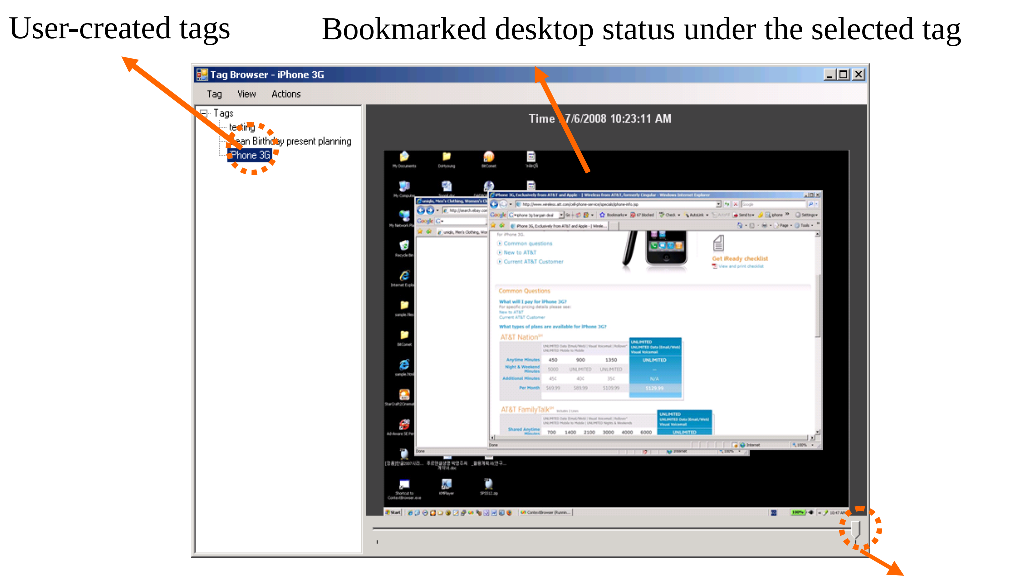


Figure 16. An example of visualized tagged and bookmarked narratives of a desktop during the selected time fame.

7.2 Tagging and Bookmarking Desktop Status

To users, browsing a timeline is analogous to scanning a document pile on the desktop in their office. How deeply users organize information is largely depending on the users' style, although a study showed that a type of job may affect the way of organizing information [Malone, 1983]. A timeline interface guarantees that users can find previously used desktop status in which they performed a task, even though they did not file away use of information at the time. The underlying hypothesis supporting the aforementioned is that the users revisit their tasks frequently, which makes a time a crucial clue to locate what they were looking for and the length of a timeline the users should browse relatively short. However, they can even skip this procedure by tagging and bookmarking desktop status that is considered critical enough for them to revisit in



a. Tag browser

b. Tag and bookmark information on the timeline

Figure 17. Tag browser interface (top) and a bookmark displayed on the timeline view interface (bottom)

the near future. We make a bookmarking feature, which is a commonly important feature for various Web browsers, available to users' desktop for the very same reason – to store important desktop status to allow them to revisit and reuse in the future. Tagging is another important feature in the current Web environment in which a tag represents a semantic value of the content of a particular Web page. Yet another facet of making a bookmark, which is eventually associated with a tag in the context browser, allows users to build a semantic structure in a time-based pile in which a time is the only structural property. By use of a tag and bookmark feature, piled narratives of a desktop start having such user-defined structural properties by which users can easily recognize and retrieve meaningful periods or discrete states of the pile, see Figure 16. The immediate beneficiaries of this feature are those who habitually organize their task-related information diligently. Users can tag and bookmark either while performing task, i.e., managing at the scene, or at users' convenience, i.e., delayed classification. Tags represent themes of and bookmarks indicate states of what they are or were doing. With tags and bookmarks in their hands, users can easily scan narratives of a desktop and retrieve desired contexts even without browsing the timeline, see Figure 17. With the *tag & bookmark manager* in the context browser, users can directly restore the task's context in which they want to resume.

As described in Figure 17, when users select a tag, the manager shows the bookmarks associated with the tag and allows them to browse those bookmarks. Launching bookmarked status is clearly the way users can restore their context to further the given task.

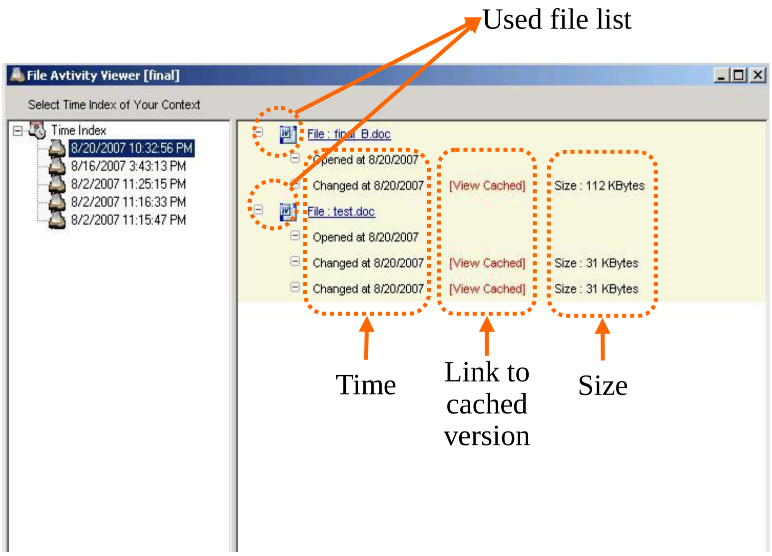


Figure 18. File activity viewer

7.3 File and Web Activity Viewers

While the timeline viewer presents abstract visual flow of a desktop status by a time, the file and Web activity viewers present concrete views toward users' file and Web activities. With the file activity viewer, they can see their task-related files and history of them in one place. As described in Figure 18, the viewer shows all used files and every status change on a corresponding file, such as when it was created, changed or deleted. Further, it tells the sizes of the file upon those changes, which may inform the users when a significant change was made by seeing the stream of changes on the size of each state – significant size change might advise that a noticeable progress or decision had been made to the file or even to the task. Using this characteristic information and the timeline viewer together, users will be able to revisit the context at the time when any significant changes had been made. Similarly, the Web activity viewer , see Figure 19, shows users' Web browsing history and it lists up the previously visited pages with the following three optional choices; 1) with a big thumbnail, 2) with a small thumbnail and 3) without a thumbnail, i.e., just URL and the page’s title. It also clusters pages whose addresses are under a same domain to keep the list from being visually complex and to help users browse the history more efficiently.

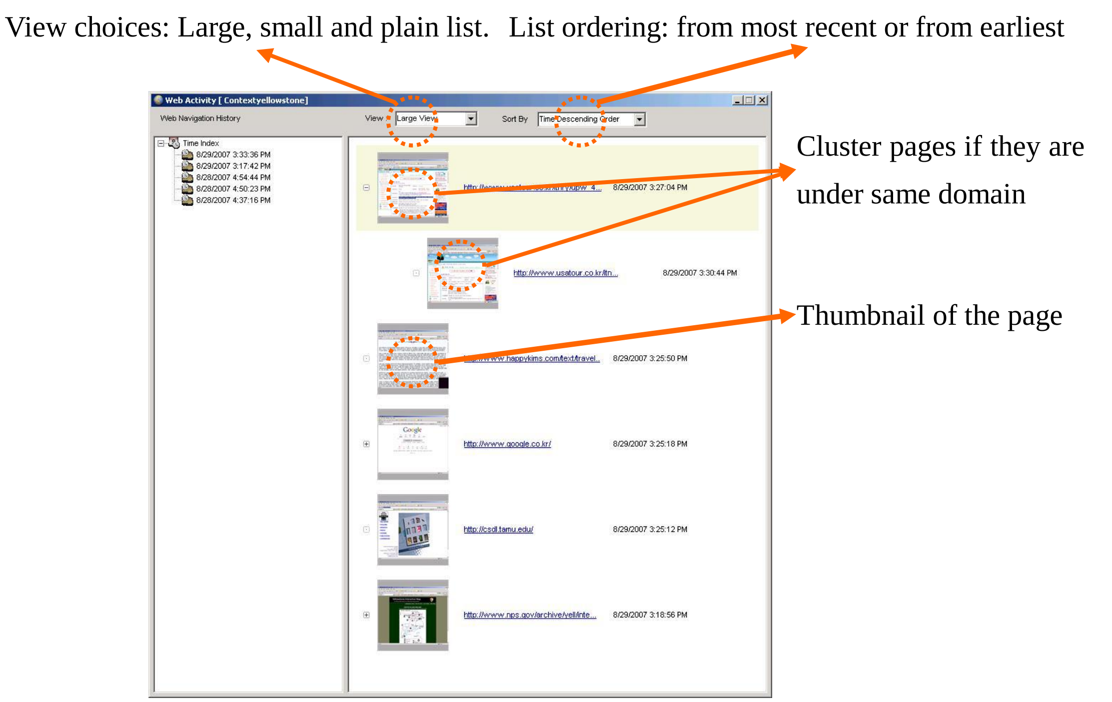


Figure 19. Web activity viewer

8 SEARCH FOR CONTEXTS

In this section, we describe the search interface of the context browser. The novel approach of the search engine of the context browser is that it searches not only for individual information units such as files, but also it delivers the contexts in which those searched items were being used. In this vein, we prefer to refer to this as context search rather than just search. As we described in a SECTION 6, the context browser has an internal search module in which there are a stemmer to generalize the terms in both the query and the corpus and a vector space retrieval engine to cosine term vectors to measure the similarity between them.

8.1 Search for Context: Desktop View

The fundamental search targets are eventually files or Web pages – actual window objects pointing to the searched files or Web pages to be exact. Based on that, we want to provide desktop status (work context) in which each searched item was being used, which lets users recognize the surroundings, i.e., associated information at the time. By providing users with not only searched results but also their contexts, we want to achieve the two goals of restoring 1) desired information and 2) users' cognitive context at the time. This provides not only a list of files or URLs but also what was going on, say, when users were writing a document.

Figure 20 shows that the *desktop view* interface of the search manager provides the desktop status of the selected item of the searched list. With this interface, users can launch not only the searched window object on a desktop view, but also they can execute any other window objects they consider related, not to mention launching whole desktop status as shown. Further, they can revisit the timeline when the searched window was active to skim narratives from around the time, which allows them to obtain more refined search results.

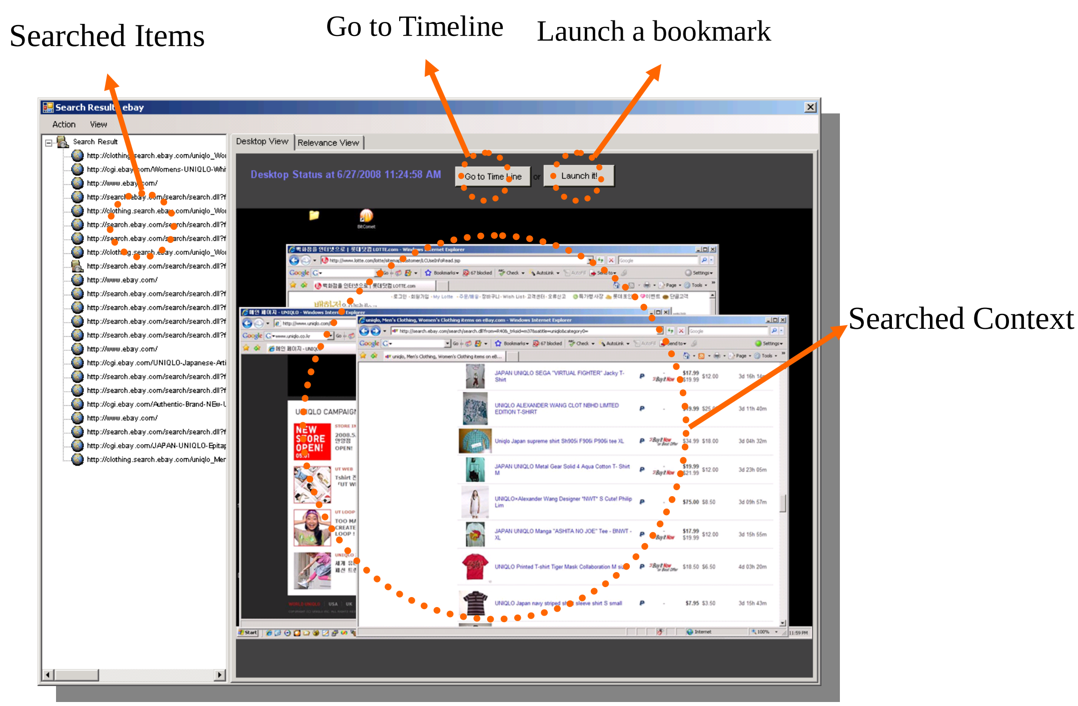


Figure 20. Dekstop view interface of a search manager

8.2 Search for Relevance: Relevance View

A salient feature of the context archiver is to maintain window switching events that have taken place during performing tasks and build a *window switching* matrix, *WS*, where ws_{ij} is proportional to the number of events that switched the focus from window w_i to window w_j. We believe that users often switch around particular windows while performing a task when the windows are semantically associated. To measure a semantic relatedness between windows, we pay attention to the switch frequencies between windows – a high switch frequency between two windows may indicate they are highly semantically related.

Figure 21 shows that the *relevance view* interface juxtaposes maybe-related

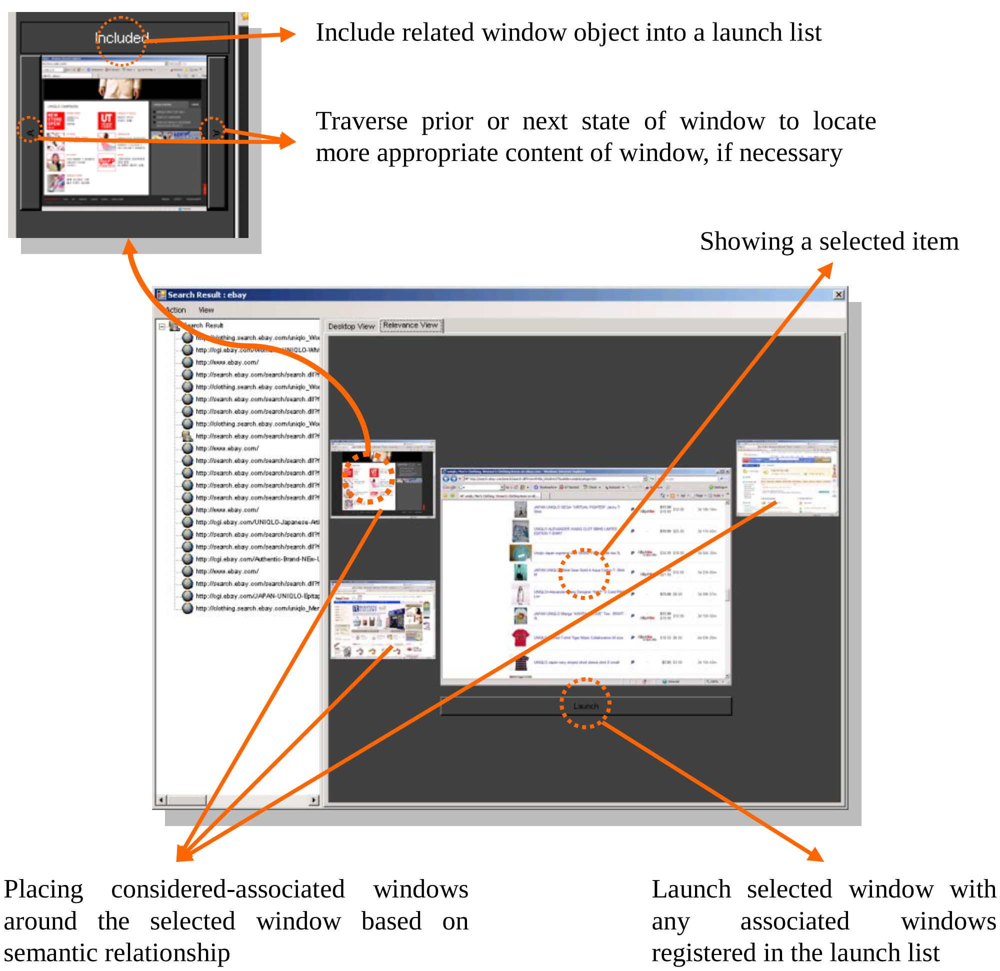


Figure 21. Relevance view interface of a search manager

windows around the selected window based on the level of relatedness. It shares the same output produced by the search module with the desktop view interface. The difference is that when users select any from the searched items, it then retrieves the windows running in parallel at the time with the selected window and places the windows according to the switch frequencies between two parties. The windows with higher frequencies are put on the left or right side of the selected window and the ones with lower frequencies are put at the top or bottom. It highlights the results in a more analyzed way so that users can quickly recognize what other resources are actually relevant to the selected one.

9 INITIAL EVALUATION

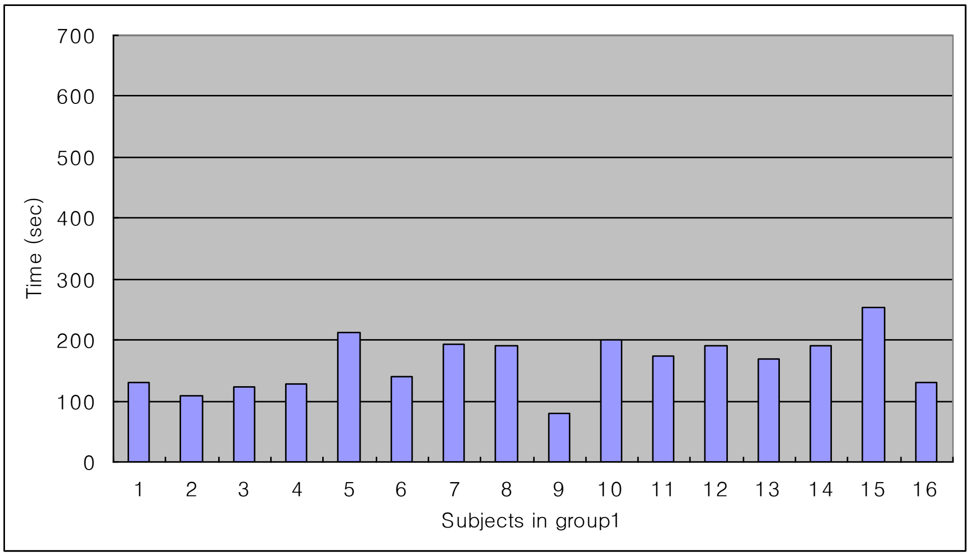
In this section, we describe the initial experiment we conducted in our lab to evaluate whether the context browser can actually help users better handle interruptions and resumptions.

9.1 Subjects

22 graduate students (21 from engineering majors and 1 from the MBA program; 4 were female) participated in the study. The reason for selecting the most from engineering majors is that we want to have heavy computer users contribute to the study. 16 subjects in *group 1 (with the context browser)* performed the task with the context browser and the other 6 subjects in *group 2 (without the context browser)* performed the task without the context browser. We assigned more subjects to group 1 since we wanted to focus on uncovering the effectiveness of the context browser.

9.2 Task and Procedure

Each subject was asked to plan a trip lasting seven days and to document the plan using a computer that we provided. Planning a trip requires a subject to gather a variety of information, such as flight ticket prices, hotel reservations, local information, special events s/he wants to participate and so forth. Each task was performed in two separate sessions and each session took approximately one hour. The period between the two sessions was about 24 ~ 48 hours. The reason for dividing each task into two separate sessions is to create an interruption for the subject.



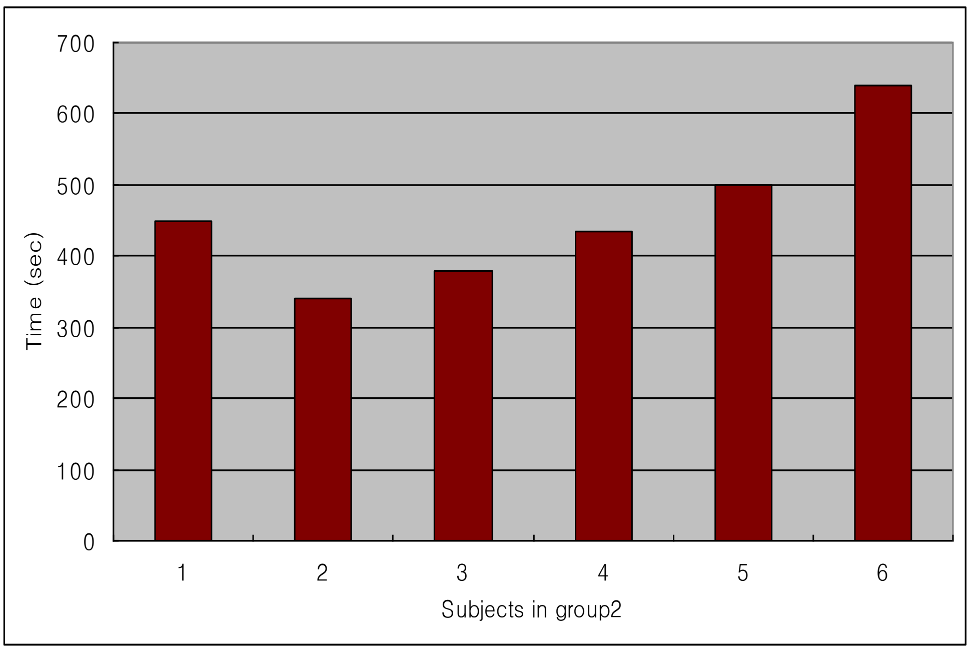


Figure 22. Task tracking time

9.3 Findings

9.3.1 Task Tracking Time

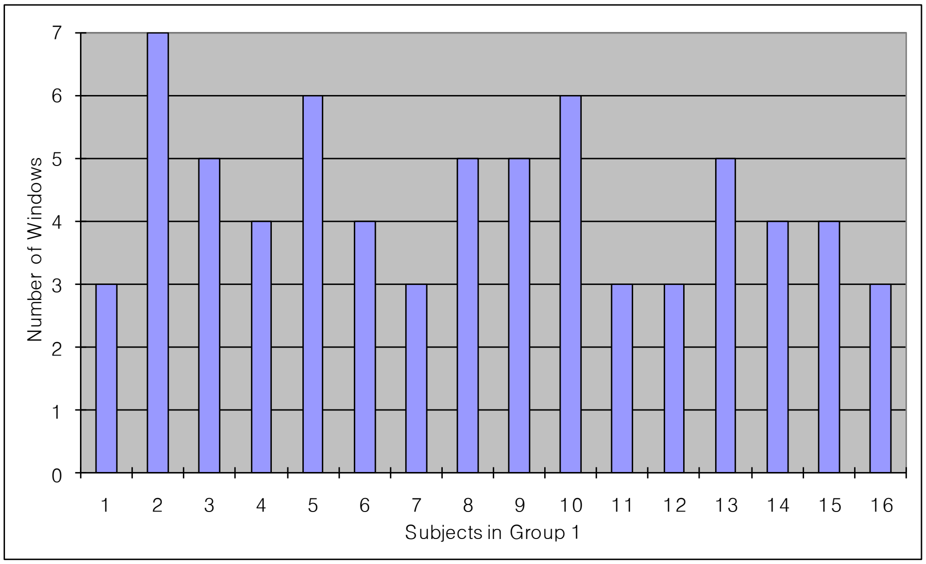
The task tracking time expresses how long it takes for a user to prepare the resumption of the task, such as figuring out what to do and use, and what kind of information was used previously. According to the study performed by Czerwinski, et al. [2004], 13% of participants' tasks were task tracking. In order to measure the task tracking time, we asked all subjects to let us know when they were ready to carry on the task. As shown in Figure 22, we can observe a trend that subjects who used the context browser (group 1) spent less time (avg = 163 sec., sd = 45) than the subjects in the other group (avg = 456 sec., sd = 102), t = -9.5, df = 20, p<0.001. Because the context browser provided most of the information regarding what a user was doing and which information was visible on a previous desktop status, it took less time for subjects in group1 to retrieve such information. Interestingly, during interviews with subjects who used the context browser, most of them expressed that exposure to the desktop status they previously bookmarked, immediately triggered recall of various things related to the task. This may enrich their performance and augment continuity of the task.

Table 1. Reasons for leaving a computer powered on

Q) What is the main reason for leaving a computer turned on?	
I don't know	**2/22 (10%)**
To save booting time	**1/22 (4%)**
To avoid losing current information	**12/22 (54%)**
To resume easily when I get back to the work	**7/22 (32%)**

9.3.2 Richness of Task Preparation

Prior to the user study, subjects answered a questionnaires about how to use a computer. One of the questions and answers are showed in table 1. The subjects indicated that “leaving the computer turned on” is used as a mechanism to keep the last status of their work, since this allows them to continue the work from exactly where they left off in their last sitting. Especially, if the previous desktop status contained many information objects, this enables a user to resume in a rich context without much cognitive burden. The richness of the task preparation thus addresses how well a user is prepared prior to continuing the task. To measure this property, we counted the number of information objects—the number of initiated windows on a desktop when a subject is ready to resume the task, i.e., when tracking a task is finished. Basically, this tells us how many information objects are actually retrieved by a subject. We believe this could indicate the level of readiness of each subject in continuing the task. Measuring explicitly how well each subject is prepared is not something that can be easily accomplished with numeric data. However, there might be a difference between when a subject continues his/her task with five or six information resources and when a subject does with one or two information resources. We can thus postulate that more information resources previously associated with the task can enrich the readiness of task readiness more than less information resources do. As shown in Figure 23, there was such a pattern in that subjects in group 1 resumed their tasks with more information objects (avg=4.3, sd=1.26) than subjects in group 2 (avg=1.6, sd=0.54) did, $t=5.35$, $df=20$, $p<0.001$. We expect that richer and more contextual information can lead a user to better performance over a task generally.



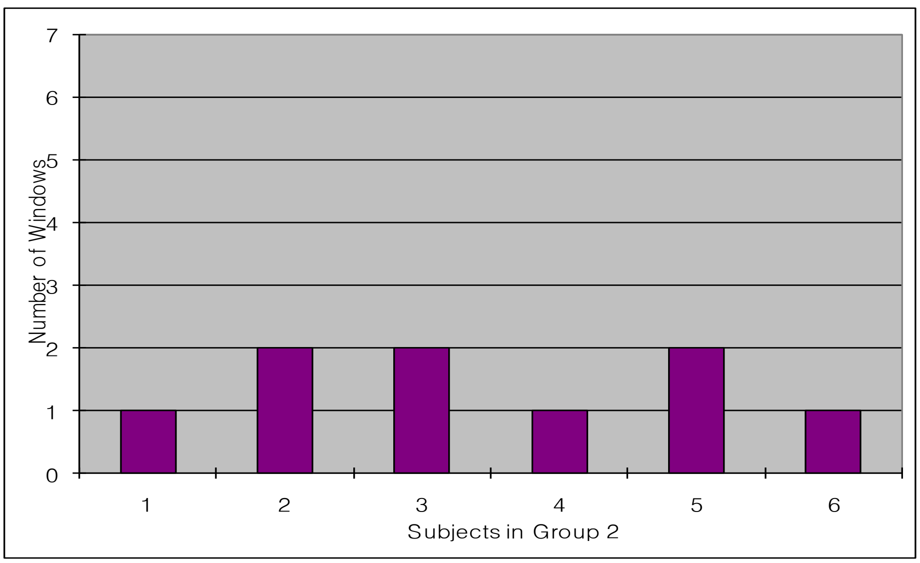


Figure 23. Number of windows when resuming the task

10 EVALUATION

In this section, we describe the user study we conducted to verify the validity of the system. The goal of this study is to discover clues for whether the system serves users' need or not.

10.1 Subjects

24 subjects (5 were female) participated in the study. They were not compensated for this participation. Ages ranged from 20 to 39. 14 were married and 6 out of 14 had a child(ren).

10.2 Design

An ideal experiment for this particular research would be testing the system under the situation in which users perform their own tasks in their actual office or working environment. However, for practical reasons, the evaluation was done under a managed short term work context (around 2~3 hours of user data) rather than for a whole project duration which can last up to several weeks or months depending on the type of project. The primary goal of the research is to augment users' work practice by minimizing their time and cognitive costs. To perform between subject measures, we divided subjects into two separate groups (12 subjects per group). Subjects in group 1 used the context browser and the others in group 2 did not use it.

10.3 Tasks

During the study, participants were placed in a simulated office environment in which they encountered frequent interruptions and performed multiple tasks in parallel or in sequence. To make such an environment, we prepared a set of tasks in advance and initially assigned three tasks to subjects prior to the study. Each task may have a different level of workload, in terms of quantity of information they had to process and mental workload. By assigning tasks beforehand, they could thus briefly develop a task-related strategy such as ordering tasks, calculating approximate time expense for each task and so forth. The following is the list of tasks that were initially assigned to subjects.

10.3.1 Task #1

Find pictures of given 9 cities. Find 5 pictures per city from the WWW. However, there are two condition subjects must meet. The conditions are 1) one picture should contain a sea or lake or river scene and 2) one picture should contain a central station of the city, except in the case the location happens to be an island. Subjects may download and save images in the folder of a corresponding city. Or they may create a document that contains the images. From now on, we refer to this task as *City pictures task*.

10.3.2 Task #2

Search for Van Gogh's famous paintings. (At least 10 paintings). Sort the paintings chronologically and research their current locations such as a museum's name and its address. Make a document that contains the images of those paintings and their current locations. Again, the data in the document should be chronologically organized by the year of each painting. From now on, we refer to this task as *Gogh task*.

10.3.3 Task #3

Subjects are about to move to San Jose, CA due to a recent job offer. Their monthly salary is $5000 after taxes. Research information on housing or apartments of the city and find at least 3 candidates to move in. From now on, we refer to this task as *San Jose task*.

Each task has different internal properties that can affect the way of task execution by subjects. The City pictures task explicitly expresses 9 subtasks, which may give an impression of heavy work load in terms of quantity. However, it does not require users to pay attention to its progress or continuity of a task's context since the task has initially 9 break points due to existence of 9 subtasks. The Gogh task is explicitly a single task but it may also be implicitly broken into 10 sub tasks all of which are regarding his famous works. In addition to that, it requires an additional documenting task and a little bit of cognitive load by asking subjects to make the paintings chronologically organized in the document. Lastly, the San Jose task is less defined and descriptive in terms of its structure and conditions, compared to the previous two tasks. Conditions given to subjects are their monthly payment and the city where they are about to move. Hence, they have to decide all kinds of variables such as "rent or sale",

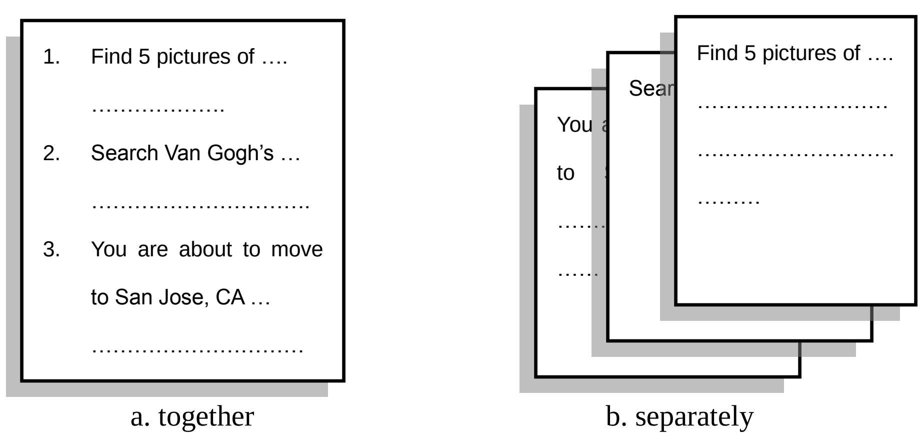


Figure 24. Two different ways of assigning tasks

"house or apartment", "school district (if they have children)", and so forth. Further, we assign tasks in two different ways, as depicted in Figure 24. Subjects were assigned tasks either by one sheet including all three tasks with task numbers or by three separate sheets each of which contained a task without task numbers. In doing so, we could observe how they order and organize tasks in two different cases. Finally, subjects were not required to complete these tasks and could carry them out in any order. However, they were responsible for making some progress for all three tasks during the first session. In the second session, subjects were asked to continue the tasks and we did assign tasks in the order that we defined.

10.4 Interruptions

In this study, we expected three types of interruptions to happen; 1) internal interruptions, 2) external interruptions and 3) disruption.

1. **Internal interruption**: It is initiated by subjects when they want to switch to a different task. By having multiple tasks on their hands, they will invoke multiple internal interruptions [20] in a self-motivated way to switch to, and initiate the next task. Since subjects were required to carry out multiple tasks, they had to suspend the current task in order to perform the next scheduled one.
2. **External interruption**: It is caused by us. To simulate an actual environment more realistically, we also prepare another set of tasks, and those were assigned during the study and were designed to cause an interruption to the subjects' current task. We interrupted them by assigning additional tasks while the subject performed tasks.

The subject should suspend the current task and carry out newly given task. After completing the task, they can resume the suspended task.

3. **Disruption**: It is an accident that takes place while subjects perform a task. Basically, the subjects lose current status because of the accident and they must restart the computer and try to resume the task. This incident compromises a document that they have been working on and is a tool to impede continuity of the task.

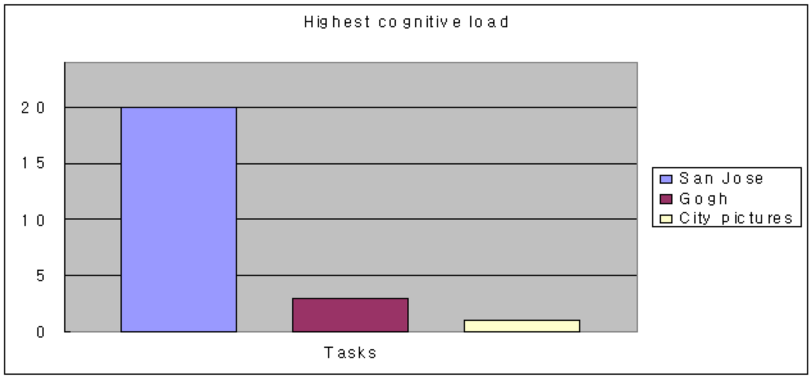
10.5 Findings

10.5.1 Cognitive Workload

After the study, we asked all subjects which task caused the highest mental workload. As described in Figure 25, they (20 out of 24) considered the San Jose task to be often associated with mentally loaded activities. The primary reason for this result is that the task required various decision making processes such as whether to live in a house or an apartment, whether to rent or buy, whether to live in downtown area or outside of the city and finally how much they were willing to pay. Choices they made to live in San Jose are described in Table.2. The married subjects expressed high cognitive burden because buying a house is more difficult than looking for an apartment.

Table 2. Different APT/House select ratio upon marital and have-a-kid status

Condition (number)	Apartment	House
With a kid(s) (6)	0/6	6/6
Without a kid (8)	3/8	5/8
Married (14)	**21% (3/14)**	**79% (11/14)**
Single (10)	**90% (9/10)**	**10% (1/10)**



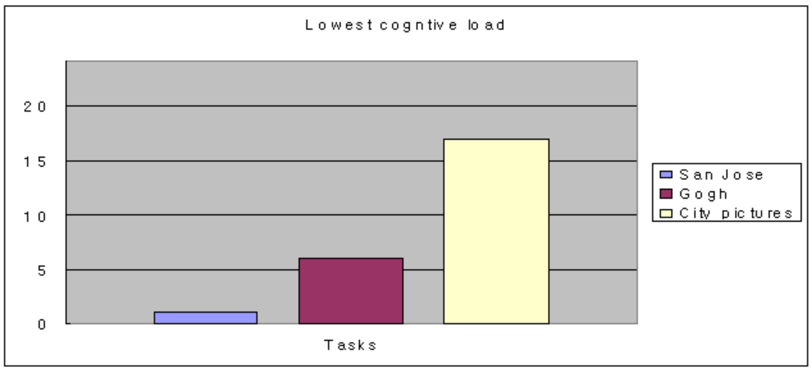


Figure 25. Ratings of workload

(top: highest workload, bottom: lowest workload)

Particularly, the subjects with a kid(s) conveyed higher workload since their first priority tended to be searching for a better educational environment for their kids. 50% (3/6) of them could not complete the task as opposed to all single subjects, who successfully completed the task by finding 3 candidates. On the other hand, the City pictures task was considered the easiest task since the task was a collection of fairly simple subtasks.

10.5.2 Ordering Tasks

As stated above, we assigned tasks in two different ways to observe how each subject orders given tasks and the result is shown in Figure 26 (task #1 is the City picture

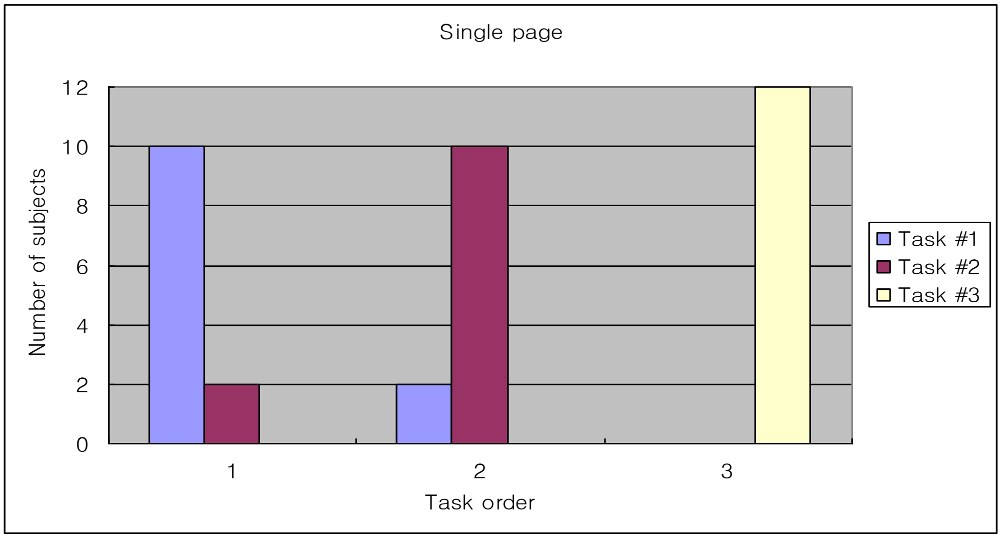


Figure 26. Task orders in two different assignments

task, task #2 is the Gogh task and task #3 is the San Jose task). The subjects who were assigned tasks with a single paper mostly (except 2 of them) followed the order of the tasks in the paper without giving much thought. Some of them actually thought the order of tasks in the paper was reasonable and followed the order. The other subjects who were given tasks with three separate sheets showed the fact that they actually tried to develop their own strategy in executing tasks. There are two major trends in ordering tasks: difficulty and quantity. The majority of the subject with three sheets decided to perform the San Jose task later since they felt that a lot of variables should be resolved to get the task done. Further, the subjects who performed the task first wanted to deal with the most difficult task first rather than postponing. Interestingly, there was no one who did the task second. Many subjects, however, performed the City picture and Gogh tasks first or second, since those two tasks were considered relatively easier in comparison with the San Jose task. More than half of them performed the City picture task first or last. They explained that the task required them to search up to 45 pictures of given 9 cities. Due to the quantity of pictures, the task was either first or last choice, which might explain that some performed the San Jose task first because of smaller quantity (only 3 house or apartment candidates were required).

10.5.3 Content Consistency

In the 2nd session, we put subjects in a problematic situation in which the document they had been working for the Gogh task was not available due to a simulated system failure, which eventually forced them to re-create the document again. By applying this problem to subjects, we tried to compare the contents of the re-created document to the ones of the original document to determine which one showed the better score of consistency and we present the result in Figure 28. The content of the document consists of images and their associated information, and in particular the name of the image file they retrieved from the WWW was often irrelevant to or insufficiently informative about the actual painting. As a result, re-creating the document eventually

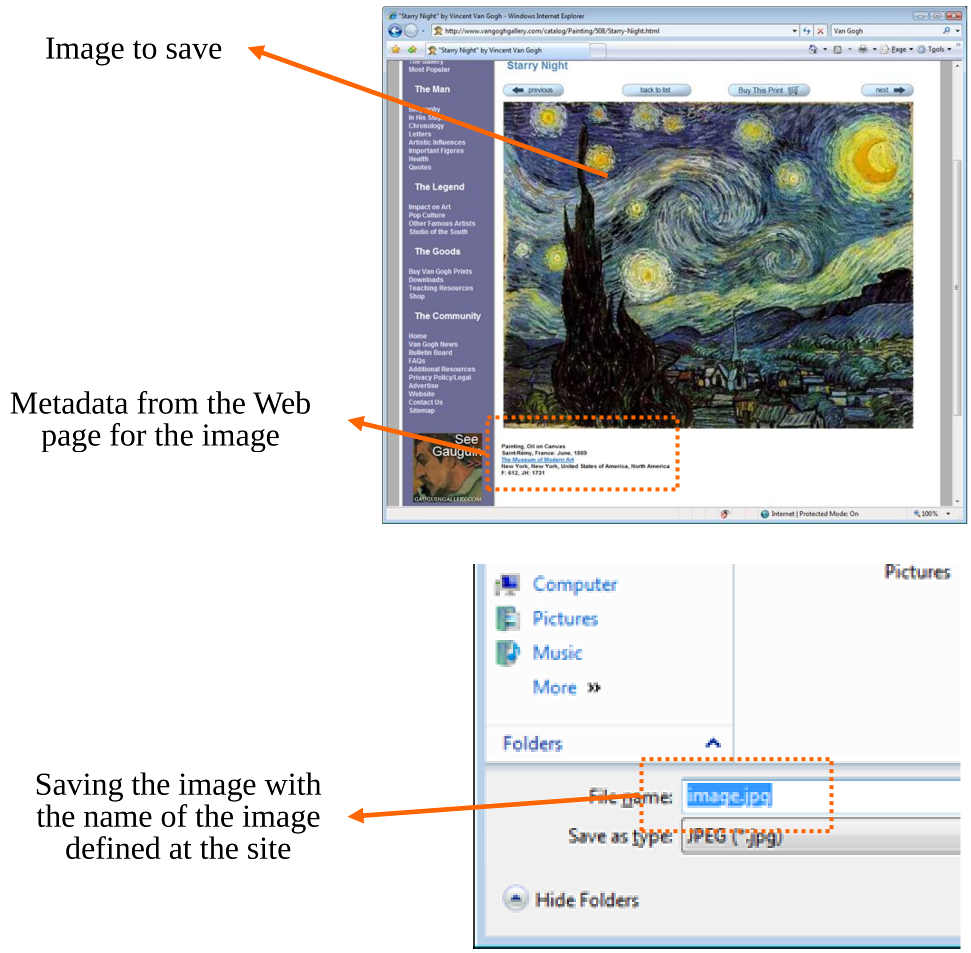


Figure 27. Making the content of the document for the Gogh task

required them to re-retrieve information on the painting at the Web page where they found the image, even though they were still retaining the images they originally downloaded in the local hard drive, see Figure 27. In an effort to gauge the difference in consistency between those documents (namely before and after), we identified how many paintings were included in the reorganized document which actually existed in the original document. The context browser has a caching feature that saves the user's

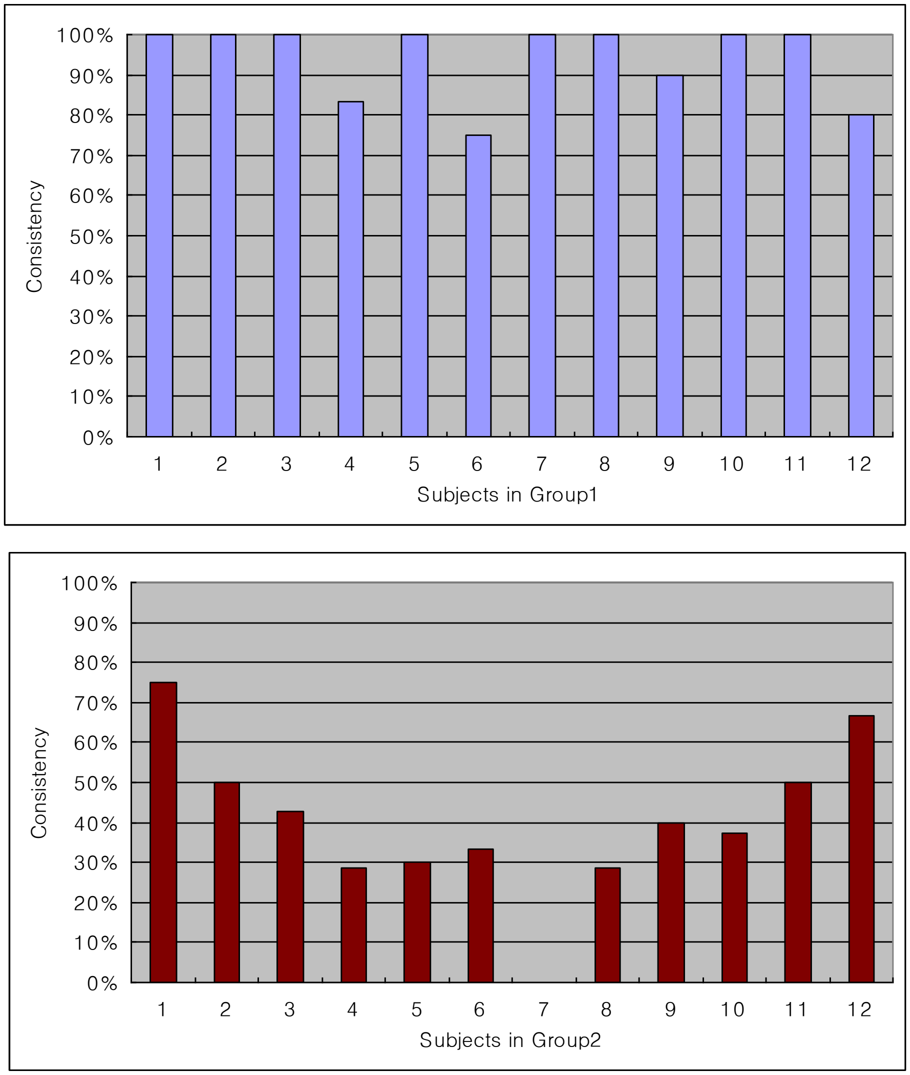
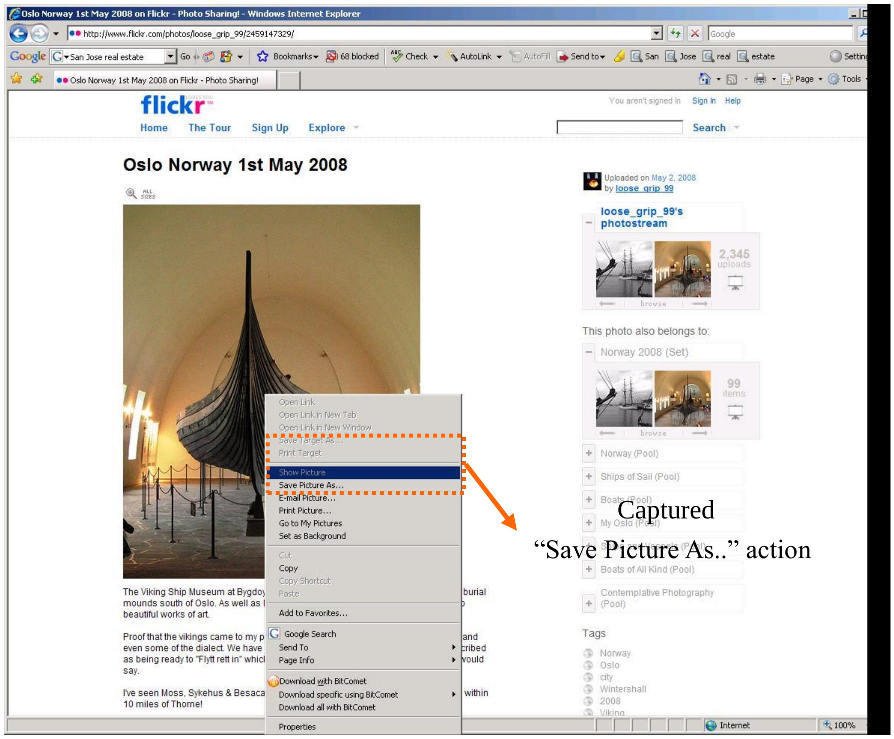


Figure 28. Content consistency (top: group 1, bottom: group 2)

working file when it reports any changes to itself. In a case that users cannot open an original file, the context browser allows them to retrieve the lastly cached version instead. Thanks to the feature, the subjects in group 1 (avg=0.94, sd=0.09) reported much higher consistency rates than ones in group 2 (avg=0.4, sd=0.19), t=8.633, df=22, p<0.0001. There was even a subject (#7) in group 2 who created a document with all new paintings. In addition, we could suspect that even without the use of a document caching feature group 1 would outperform group 2, because they could still get a better chance to identify which painting was selected from which Web page containing the information on the painting using the timeline interface.

10.5.4 Recall and Precision

The city pictures task echoes such tasks that are associated with a considerable amount of visually similar information resources, such as graphs and images. In reality, there might be the situation in which users need to retrieve and review all resources they had already seen previously either to verify that they have a proper result or modify the current result. To simulate this situation, in the 2nd session we additionally asked subjects to do the following – "Given 10 minutes, find the images that you believe that you saved while performing this task." Figure 29 shows useful cues that subjects were able to find useful, and Figure 30 and Figure 31 show the charts for recall and precision rates for the re-retrieved images by the subjects in both groups. This measure expresses when a task requires so diverse information resources and users need to track down those resources used earlier, how they perform with and without a proper aid. In general, group 1 (avg=0.643, sd=0.03) shows better recall and precision rates than group 2 (avg=0.4, sd=0.14), t=5.45, df=22. $p<0.0001$. The subjects in group 1 explained that with the context browser (the timeline view interface to be exact) they were able to chronologically track down most of the Web pages where they found and saved images previously without any significant difficulty. In particular, the subjects from group 1 managed to find additional contextual cues besides the piled narratives of a desktop, as seen in Figure 29, from the use of the context browser. The cue from the reconstructed

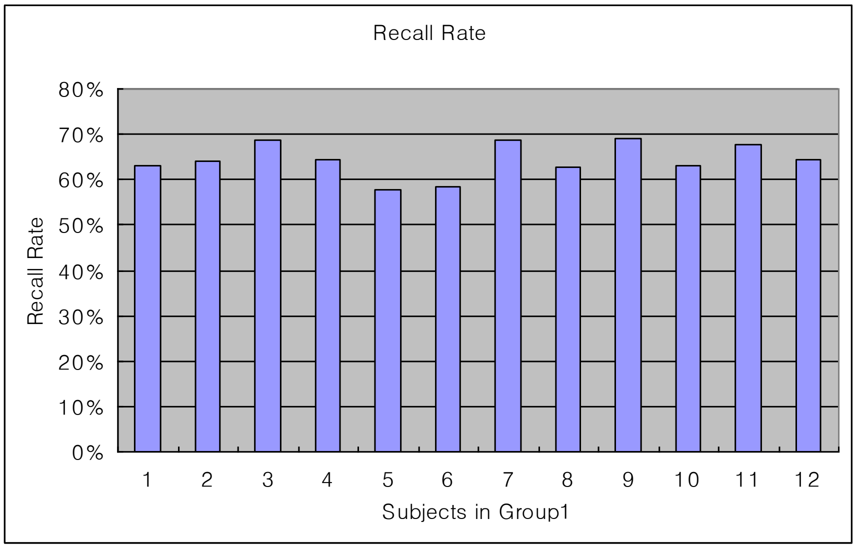


a. A cue from a desktop

b. A cue from the timeline

Figure 29. Additional contextual cues subjects found useful

desktop, see Figure 29-a, often directly informed them of which image was saved from which Web page by exposing the very act of “Save Picture As…” performed on the Web browser. They also found another cue from the timeline, which was file activity red-colored dots on the timeline (see Figure 29-b) by which they were able to locate the desktop status when they saved an image from the WWW. Thanks to the cues from the context browser, group 1 found more images and found them more correctly. Particularly, the difference in precision is more obvious than the one in recall. By having the subjects in group 1 exposed to narratives of prior desktop status, group 1 (avg=0.933, sd=0.46) collect more images with much higher precision within a limited period than group 2 (avg=0.59, sd=0.15) did, $t=7.23$, $df=22$, $p<0.0001$. However, the subjects in group 1 experienced a little bit of difficulty to control the timeline bar since they worked with many images during a relatively short period, which forced them to move around the timeline bar in a very finely-tuned way to avoid skipping any images they originally saved.



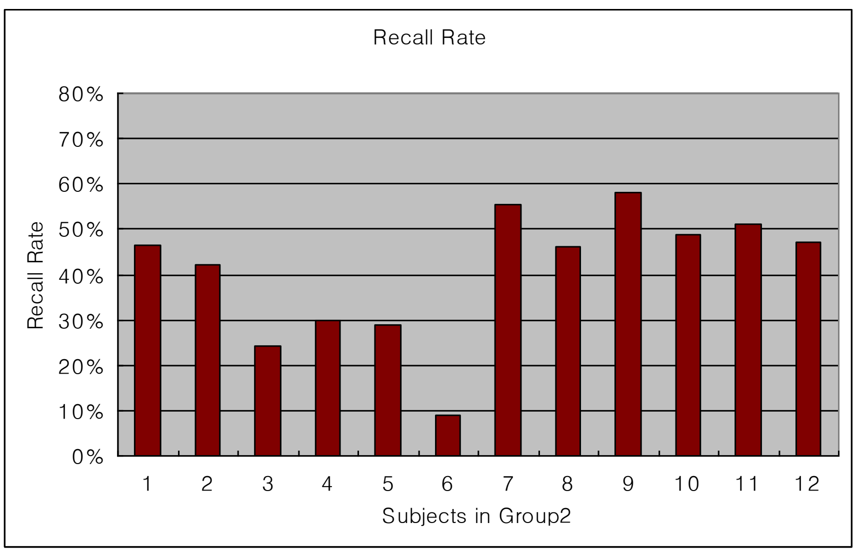
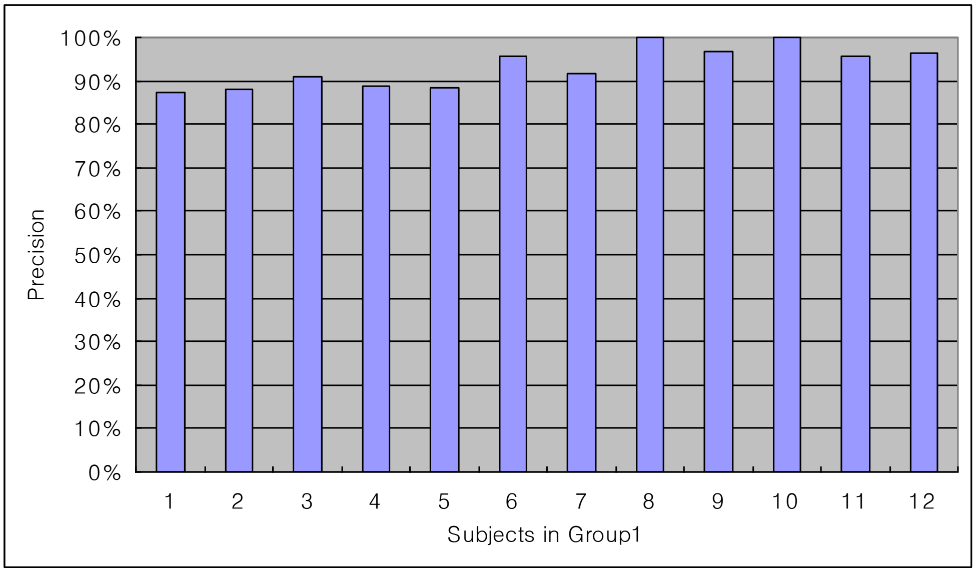


Figure 30. Recall rates (top: group 1, bottom: group 2)



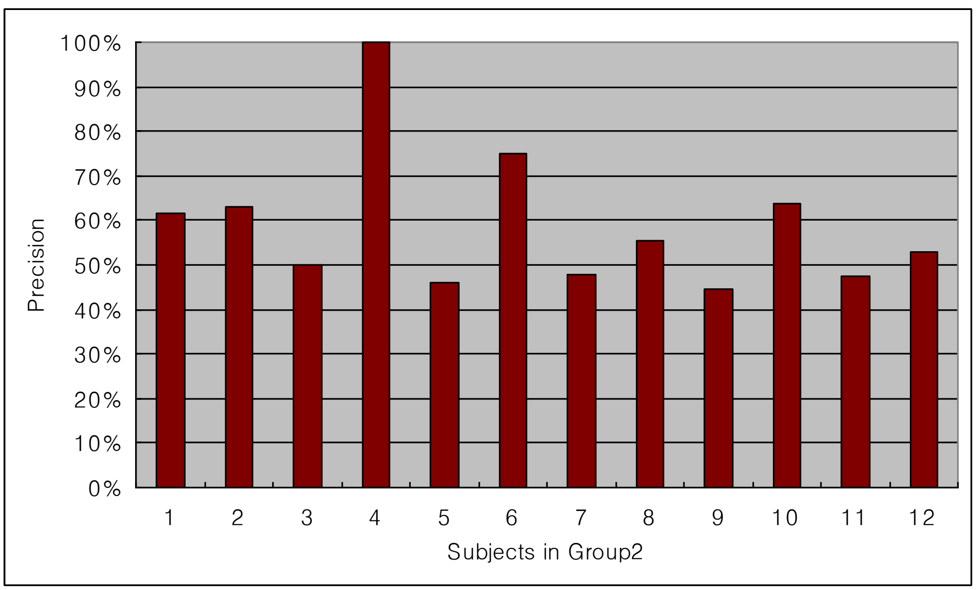


Figure 31. Precision (top: group 1, bottom: group2)

10.5.5 Time Lag to Recover Context

To find a place to live in San Jose, CA, the task forced subjects to execute cognitively loaded activities, i.e., repeated search activities, to narrow down their search list to a manageable size. Until then they have to resolve various parameters, such as a choice between house and apartment, a number of bedrooms, prices, regions, school districts and so forth. With the search list as a result of repeated search activities using aforementioned parameter, they could start reviewing houses or apartments in the list to see if there were some places interesting enough for them to put in their wish list. In the 2nd session, after they completed or were asked to finish the San Jose task, we asked them to look for 2 more houses or apartments sharing similar specification of the one they liked most among 3 candidates. The charts in the Figure 32 show time costs of both groups in finding those. Essentially, the aforementioned task asked subjects to recover previous search contexts in which they found the places they would like to live in. In terms of a time cost to restore a desired search list, group 1 (avg=245 sec., sd=70) successfully retrieved the context faster than group 2 (avg=521 sec., sd=225), $t=-4.06$, $df=22$, $p<0.001$. The subjects in group 1 found the context browser useful to get them back to the state where they were evaluating houses or apartments on the list which was the fruit of various search activities of deciding the values of all necessary variables as mentioned above. However, the subjects in group 2 again had to go through all the steps that they took previously to find them, which sometimes painfully cost a long period to execute the given task. They expressed that re-entering the necessary parameters and searching the very list which originally included the one that they originally chose were

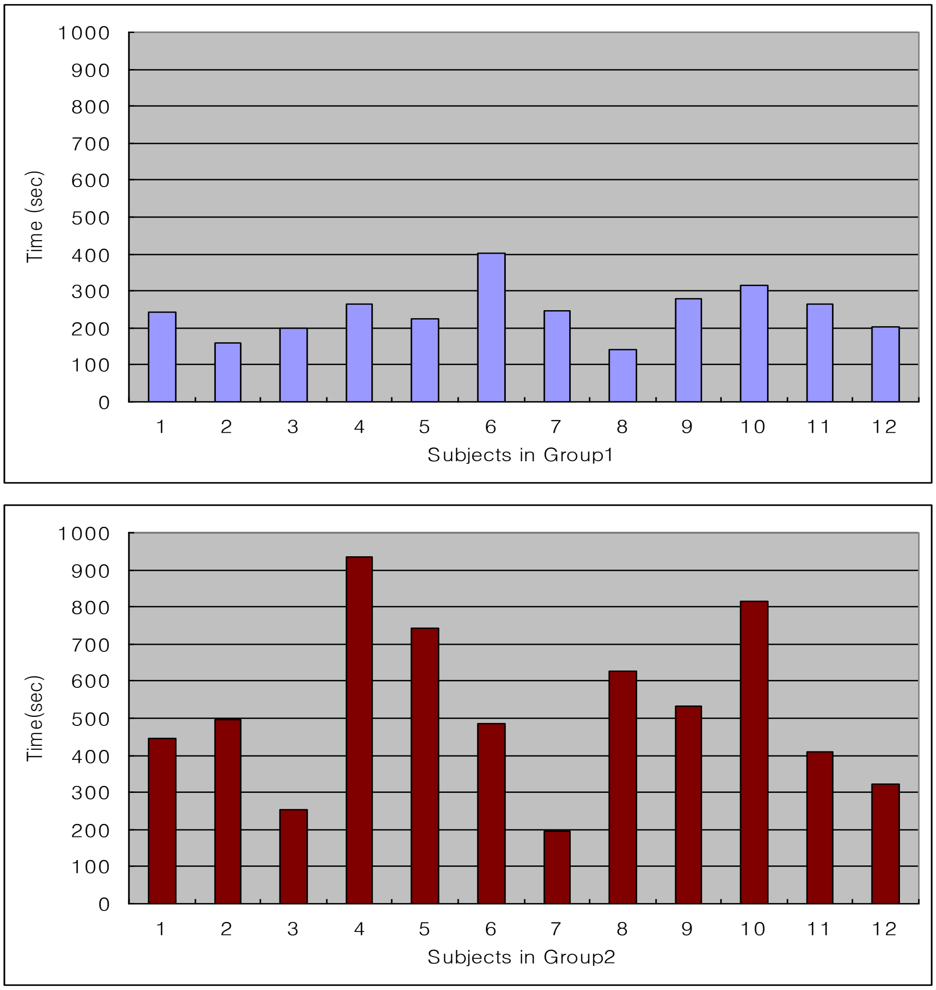


Figure 32. Time to find similar ones (top: group 1, bottom: group 2)

not enjoyable.

10.5.6 Number of Windows on a Desktop

A significant task is usually associated with more information resources, which makes users hard to revisit the task later. In general, a number of windows on a desktop may indicate the complexity of a user's current task. To discover the relationship between a task type and a number of information resources on a desktop, we analyzed

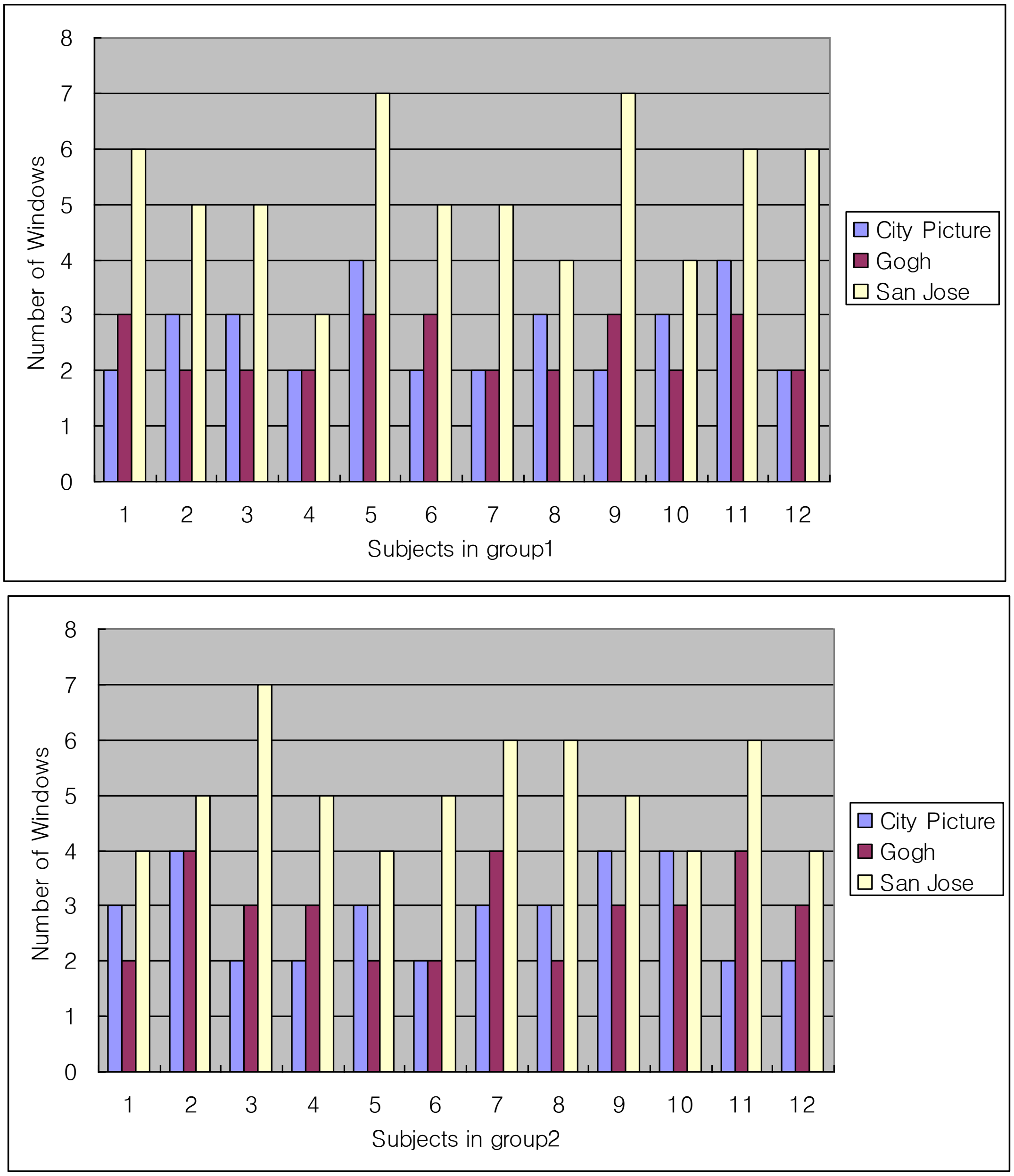


Figure 33. Total number of windows used for the tasks during two sessions

(top: group 1, bottom: group 2)

how many window objects were opened during which task. Figure 33 shows the numbers of window objects subjects used across the tasks. Regardless of subjects with or without the context browser, we learned that the San Jose task consumed most windows among three tasks. To find city pictures and Van Gogh's paintings, subjects successfully found and then heavily used a particular Web site, e.g., images.google.com and flickr.com, that allowed finding various pictures by entering different keywords or contained whole list of Van Gogh's work. As a result, they ended up using 2 or 3 window objects to perform those tasks. But, as mentioned in earlier section, searching for places to live asked them to consider various properties depending on their circumstance such as married or single, child or non-child and such. To have those conditions considered, the subjects tended to use more diverse information resources compared to the other tasks, which resulted in having more windows on a desktop.

10.5.7 Using Bookmark Tools

Marking Web pages users might want to revisit in the future has been very common to navigating the Internet, which is also available in the context browser in a more extended form. We analyzed use of bookmark features supported by both a Web browser and the context browser to discover which bookmark feature subjects decided to

Table 3. Number of bookmarks by the context browser (CB) and a Web browser (WB)

	City Picture	Gogh	San Jose
Bookmark by CB (**group1**)	3	4	19
Bookmark by WB (**group1**)	3	6	5
Bookmark by WB (**group2**)	4	13	28

use for their needs. During the study, we let the subjects decide what to use to bookmark what they need and did not encourage in any way the subjects in group 1 to use the bookmark feature in the context browser. Table 3 shows the distribution of bookmarks. In general the San Jose task produced most bookmarks among tasks – the number of bookmarks for group 1 is the sum of bookmarks from Web browsers and the context browser and the number of bookmarks for group 2 solely represents the Web browser's bookmarks. For the subjects in group 1 who used the context browser, the number of bookmarks generated by the context browser was higher than that by a Web browser. In particular, majority of bookmarks generated by the context browser had been created for the San Jose task. The bookmarks in a Web browser had been produced mostly while they were performing the Gogh task since they need to bookmark the Web site where all of Gogh's works was listed in chronological order, see Figure 34.

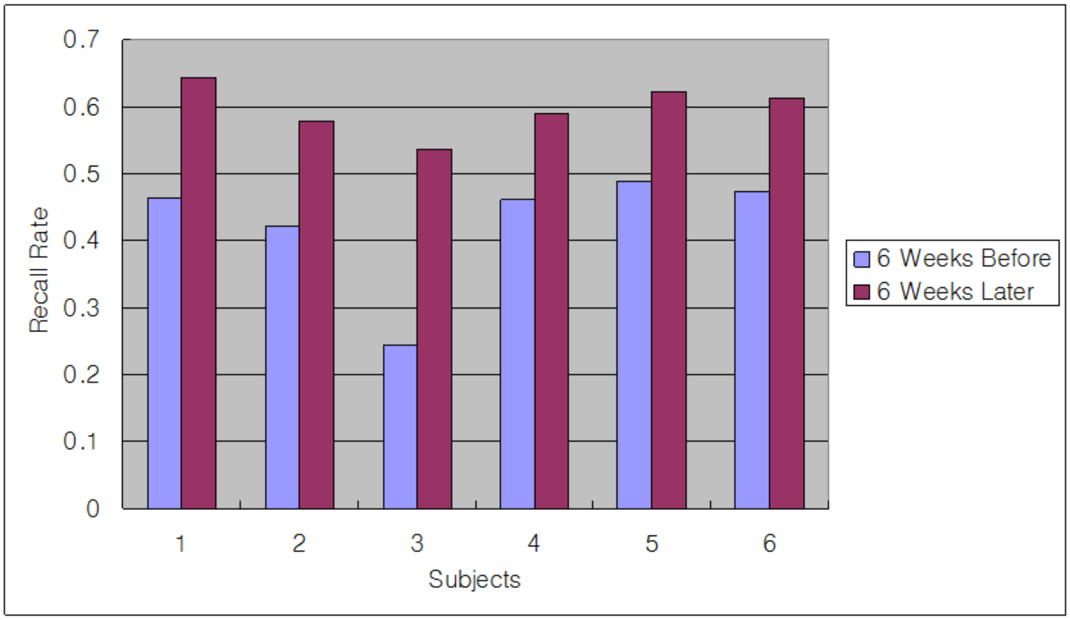
We naturally expected that the number of Web browsers' bookmarks of group 1 would be less than that of group 2, because group 1 had one more tool to bookmark than group 2 did. Yet there was a reason why the subjects in group 1 created more bookmarks with the context browser particularly for the San Jose task than with the Web browser. To search places to live in, subjects had to consider various factors and visit different real estate service sites to find a better price and place, and we claimed earlier that the subjects have more window objects on a desktop. In such a circumstance, the subjects in group 1 decided to use the bookmark tool in the context browser to retain current status rather individually bookmarking (using the Web browser's bookmark feature) all relevant Web pages or information resources opened in different windows at the time.

Figure 34. Most referenced Web sites for the Gogh task

Therefore, the subjects in group 1 wanted and hoped to retain all currently-considered factors on diverse window objects, which helps them resume the task seamlessly without considering and searching those factors again.

10.5.8 Task Performance after a Lengthy Period

We conducted one additional experiment in which we re-invited 6 subjects from group 2 6 weeks after the previous user study. We let them use a context browser at this time to see if there is any improvement in handling similar problems that they did previously even after more than a month delay. We introduced our system to the subjects and briefly explained how the system works and how they can interact with it. Since we kept a context browser running for both groups, the system was able to keep all contextual archives even for the subjects in group 2 who were not aware of the existence of our system at the time. During this additional session, we again assigned the same protocols, i.e., finding the images that you saved previously and a similar house or apartment, and measured just like we did 6 weeks earlier. Figure 35 shows the recall and precision rates of images the subjects retrieved. In comparison with their previous performance (see Figure 30 and Figure 31), we performed a paired-samples T test and discovered that with the context browser they were able to retrieve more images, i.e., a better recall, (mean difference=-0.17, $t=-6.7$, $df=5$, $p<0.001$) and the images more correctly, i.e., a better precision, (mean difference=-0.218, $t=-8.26$, $df=5$, $p<0.001$) even after the 6 week delay. They also easily found the house or apartment they liked and successfully searched similar places less stressfully and more quickly (mean difference=177.16, $t=3.47$, $df=5$, $p=0.018$), see Figure 36. Actually, using the context browser they easily recognized how they picked and narrowed down candidates and how they browsed the Internet to get a proper search result in which many worth-looking places were listed.



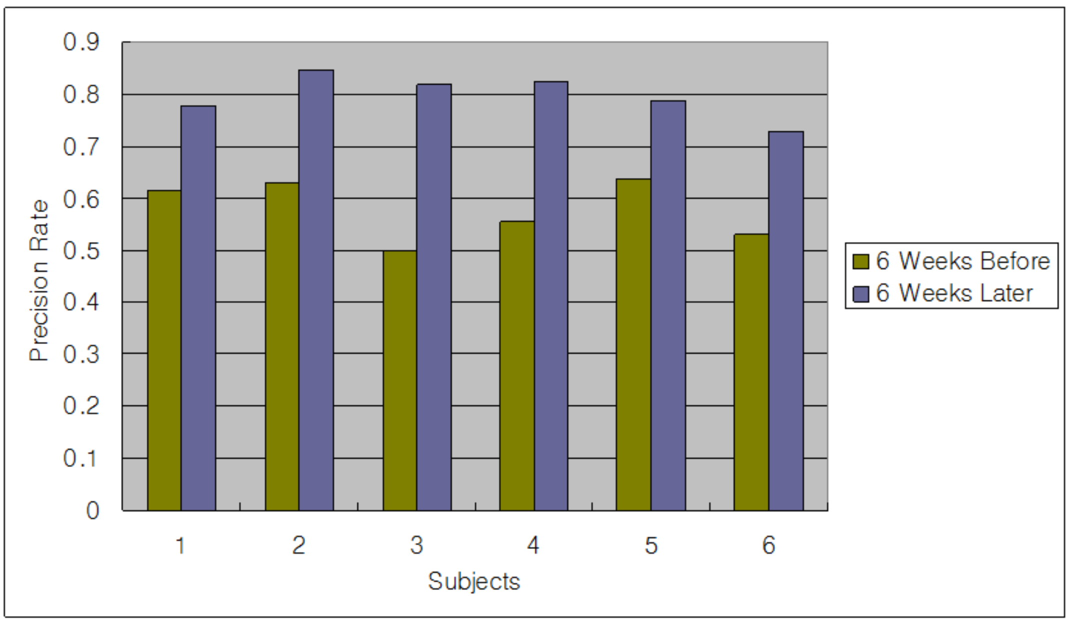


Figure 35. Recall (top) and precision (bottom) rates 6 weeks before and after (without and with a context browser, respectively)

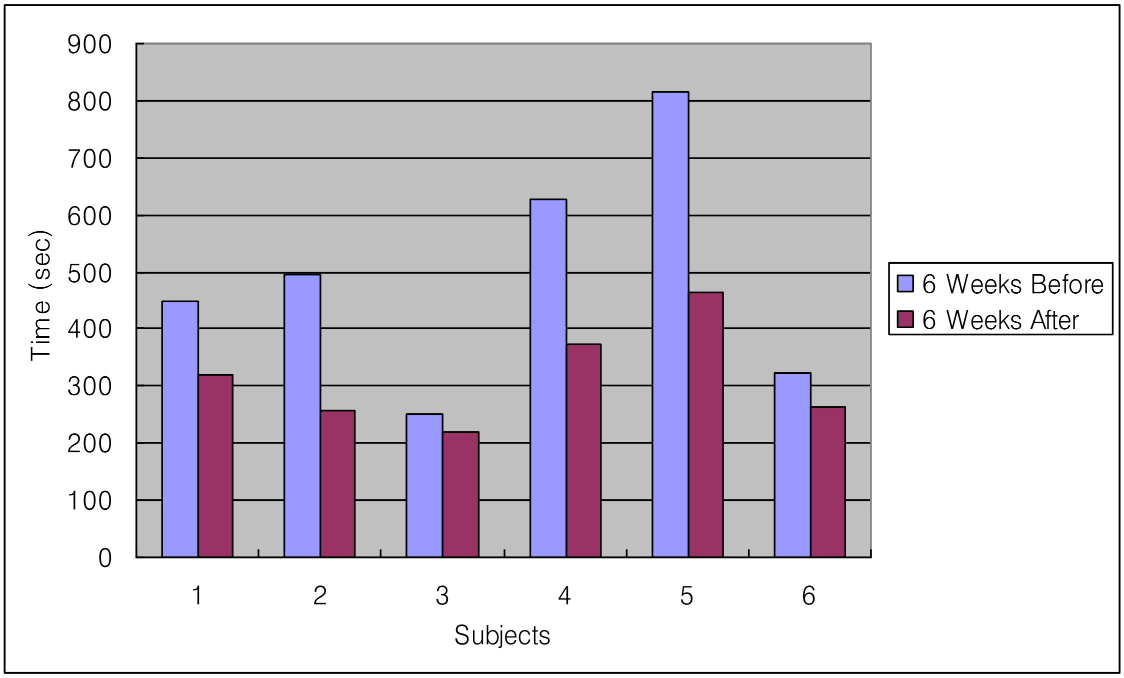


Figure 36. Time to find similar ones 6 weeks before and after

10.6 Findings from Surveys

In this section, we report several subjective ratings from our participants for the questions we asked after the study.

10.6.1 Most Effective Cue in a Short Term Project and a Long Term Project

After the study, we asked all subjects who used the context browser which feature provided the most effective contextual cue to them. 8 (out of 12) subjects answered that the timeline viewer was the one that played a critical role in helping them recover desired information. The rest, 4 subjects, answered that a combination of the timeline viewer and the tag and bookmark feature was very helpful to deal with interruptions and resumptions. The study itself was considered a short term project since it lasted only for two days. Subjects thought that browsing a timeline was adequate enough for them to find contexts since it provided them with both a time and visual cues (desktop status). Some subjects evaluated that the timeline feature with the tag and bookmark feature on the side affect the users' task performance more positively. However, as shown in Figure 37, the search feature was not considered as their first choice, which we expected. Without using the search interface, they all were able to successfully and quickly restore their desired context in a short period context. The tag and bookmark feature alone was not regarded as a strong contender, since browsing a timeline for recent activities was easy and informative enough that they did not bother to invoke the tag and bookmark manager to manage and retrieve what they had done recently.

Additionally, see Figure 38, six subjects considered the tag and bookmark feature

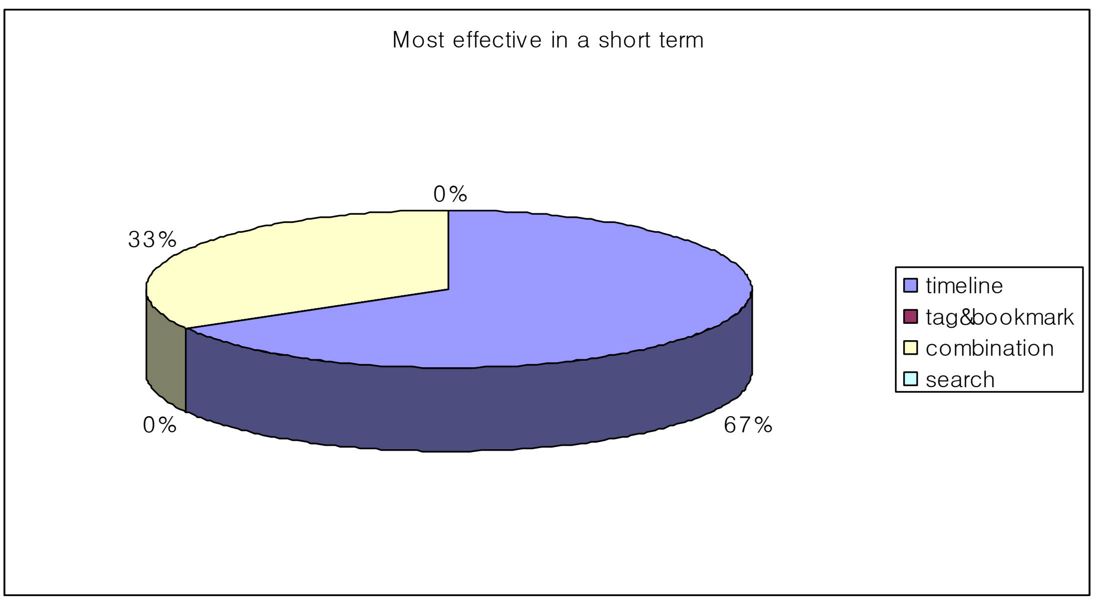


Figure 37. Most effective contextual cue for a short tem project

a most effective one, when it comes to a long term context. They expected that the feature would be useful for them to jump around specific points on a long timeline. Four of them answered a combination of the timeline viewer and the tag and bookmark feature because of the benefits of easy access both to recent moments (with a timeline) and to their important moments. During the project, there will not only be cases that sometimes they need to access the very last moment of the project, but also be ones that they need to revisit any predefined states to resolve some issues. The rest answered the search interface due to the efficiency to find information from long time ago, instead of browsing such a long timeline.

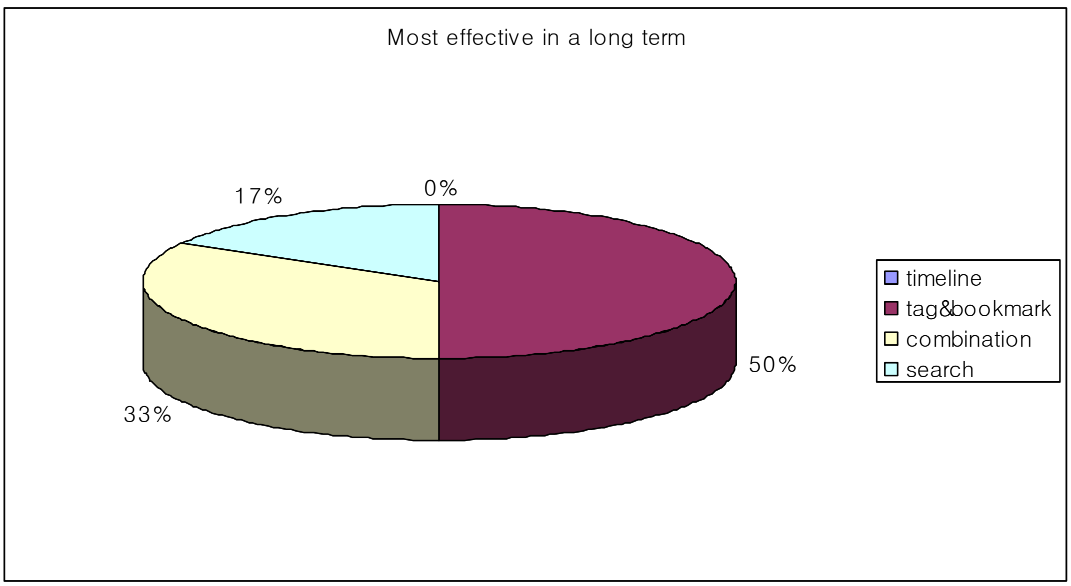


Figure 38. Most effective contextual cue for a long term project

10.6.2 Difficulty in Restoring a Document and Re-retrieving Images

We asked all participants in both groups about how difficult it was to restore a Van Gogh document and to re-retrieve the images previously saved by them; see Table. 4.

For group 1, subjects did not express any particular difficulty to perform those tasks thanks to the context browser. However, searching the images again forced them to micro-browse the timeline in order to traverse every search activity without missing one, which caused them to rate the second task harder than the first one. Other than that, all subjects in group 1 adequately managed the problematic situations using the context browser. For group 2, the subjects were able to recreate a document without any considerable difficulty once they found the Web site where all Van Gogh's paintings were listed. Yet, they found retrieving the image again extremely difficult since, besides the fact that they had to repeatedly figure out which one they saved from the search list, the poor level of their concentration on the task negatively affected their retrieval performance.

Table 4. Difficulty to restore a document and re-retrieve the images

Scale: 5(very difficult) ~ 1(very easy)

	Restoring a document Average (standard dev)	Re-retrieving the images Average (standard dev)
Group 1	1 (0)	1.6 (0.65)
Group 2	2.5 (0.9)	3.9 (0.67)

11 FUTURE WORK

We have shown that users are better able to remember and recognize semantic information presented on our system than on a standard desktop without a context browser. We believe this is because the contextual cue that a context browser is providing – desktop status, time, tag, bookmark and search – are proving helpful in retrieving desired contexts.

We also paid attention to the fact that many users had multiple computers possibly in their office (as a primary computer) and at their home (as a personal computer), which becomes another factor causing severe discontinuity. When users want to continue their task using a different computer, they again need to set up a proper task environment used in the other computer. Therefore, to make their contexts available across their multiple computers, allowing the users to export to, and import from, a different computer is a desirable approach. Consider an example that when users want to purchase a camcorder, they usually visit and search many different on-line market places to find a better deal and product. If the users cannot decide what and where to buy at the moment due to uncertainty of buying a right product, then they may want to discuss with their spouse at home. To discuss about it, they need to show what they have searched thus far, which often ends up having them spend a lengthy period of time to collect the items seen previously. However, by use of an import and export features, they just simply export what they have done to any USB flash drive and import it from the USB drive later into the current computer. Further, as mobile devices, e.g., smartphones in particular, have been matured enough to be considered computing devices rather than communication devices. Extending users' workspace to their mobile devices is a desirable approach for a certain business domain, in order to help them handle time-sensitive information quickly and appropriately.

One of the most quotidian work context in an actual office setting is a team project in which more than one team members are working together to solve a given problem. To facilitate a collaborative task environment, sharing their work contexts with each other can be a very efficient way of working together. Since with use of the context browser users will be able to share whole narrative of their desktop at any time, a sharing session does not have to be live or is not limited to any designated application which features sharing documents or spreadsheets. Therefore, if one of team members is experiencing a difficult problem, then by sharing his/her work contexts via the context browser they can easily make collective efforts to resolve the issue.

We further hope to examine the effects that the cues provided by a context browse have on the users' task performance in a long term period rather than a couple of days long project. Subject in our study presented in the previous section performed multiple tasks in two days, which also limited the subjects to using only easily perceivable features in such a short period such as a timeline interface and a tag manager. Therefore, we would like to extend our study to evaluate contributions of a context browser to a long-term project as opposed to a short one. In doing so, we want to discover how the other features such as an input density graph and a search interface which were not used frequently at the previous experiment help overall their task performance.

12 CONCLUSION

We have studied how information workers organize their task-related information, how we can make such organizing activities less cognitively-loaded and time-consuming ones, and how to support them to interact with a desktop intuitively to restore an appropriate context fast and resume an interrupted task with fewer time lags. We provided a time-base pile as a fundamental information framework for users' contextual archive since we prioritized the very benefit of the pile metaphor, which is a casual and less stressful information organizing scheme, and the known effective recall cue of time. While an automated time-based pile allows users to scan narratives in a recent period without their intervention, we need to support the users to build more structured archive by embedding additional user-defined structural properties other than just the single time dimension. To promote the chance for the users to recognize and organize their contexts, possibly faster, we therefore let them create, assign their own semantic values, i.e., tags, and mark the states for the future revisits, i.e., bookmarks. We also provided a search feature to cope with the situation in which users had a hard time to find information that they used previously even with the aforementioned features. With the search feature, it delivered the result in two different fashions; a desktop view conveying the context in which a search item was being used and a relevance view providing a searched item with semantically related ones at the time.

From the initial study in which subjects conducted a single task and experienced an interruption for a period of 24~48 hours, we concluded that the subjects with use of the context browser performed faster task resumption and retrieved more task-related information resources than the ones without use of the context browser.

From the extended user study conducted after the initial study, we were able to report more extensive positive effects on the subjects' task performance. According to the results, it showed that the subjects using the context browser showed the better recall and precision rates in such a circumstance that they had to review and re-search a large amount of previously seen information. It also reported that they were able to restore their previous search result more quickly to find the houses or apartments that share similar specification with the originally chosen one. From the interviews with the subjects, they all agreed that the timeline interface provided very effective contextual cues for a short-term project and possibly a long-term project as well. By blended use of tag and bookmark plus the timeline features, they recognized that it would be very helpful for the tasks that are lasting for a long period.

The primary objective of this research, like other research, is serving users' needs. The users, e.g., information workers in our research context, often interleave multiple tasks in a day, which gave the motivation of "reserving those different work contexts and making them easily retrievable." We implemented a system providing interfaces for users to browse and manage the narratives of their work contexts. In the course of verifying the evaluation of the system, we obtained a set of positive numbers, i.e., faster task resumption, more documents retrieved, more documents correctly retrieved and a shorter time lag to retrieve information requiring a lot of cognitive processes. In addition to that, the subjects expressed that they found the context browser very useful since it handled a tedious information organization task and kept their desired information within a close distance asking less both time and cognitive costs to find it. With both quantitative and qualitative measures, we conclude that the context browser application serves users' needs in an appropriate manner and the users also appreciate the value of the system after interacting with the context browser.

REFERENCES

Adamczyk P.D. and Bailey, B.P. (2004). If not now, when?: The effects of interruption at different moments within task execution. In *Proceedings of CHI*, Vienna, Austria, April 24-29, 2004, 271-278.

Aizawa. (2000). The feature quantity: An information theoretic perspective of tfidf-like measures. In *Proceedings of Information Retrieval*, Athens, Greece, July 24-28, 104-111.

Bailey, B., Konstan, J. A. and Carlis, J.V. (2001). The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proceedings of INTERACT*, Tokyo, Japan, 593-501.

Bannon, L., Cypher, A., Greenspan, S. and Monty, M.L. (1983). Evaluation and analysis of users' activity organization. In *Proceedings of CHI*, Boston, Massachusetts, USA, December 12-15, 1983, 54-57.

Bardram, J.E., Bunde-Pedersen, J. and Soegaard, M. (2006). Support for activity-based computing in a personal computing operating system. In *Proceedings of CHI*, Montréal, Québec, Canada, April 22-27, 2006, 211-220.

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

Blanc-Brude, T. and Scapin, D. (2007). What do people recall about their documents Implications for desktop search tools. In *Proceedings of Intelligent User Interface*, Honolulu, Hawaii, USA, January 28-31, 2007, 102-111.

Czerwinksi, M., Dumais, S, Robertson, G., Dziadosz, S., Tiernan, S. and Dantzich, M.V. (1999). Visualizing implicit queries for information management and retrieval. In *Proceedings of CHI*, Pittsburgh, Pennsylvania, USA, May 15-20, 1999, 560-567.

Czerwinski, M. and Horvitz, E. (2002). An investigation of memory for daily computing events. In *Proceedings of HCI*, London, England, 230-245.

Czerwinski, M., Horvitz, E. and Wilhite, S. (2004). A diary study of task switching and interruptions. In *Proceedings of CHI*, Vienna, Austria, April 24-29, 2004, 175-182.

Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R. and Robbins, D. (2003). Stuff I've seen: A system for personal information retrieval and re-use. In *Proceedings of Information Retrieval*, Toronto, Canada, July 28- August 1, 2003, 72-79.

Dumais, S. and Landauer, T.K. (1983). Using examples to describe categories. In *Proceedings of CHI*, Boston, Massachusetts, USA, December 12-15, 1983, 112-115

Fass, A.M., Forlizzi, J. and Pausch, R. (2002). MessyDesk and MessyBoard: Two designs inspired by the goal of improving human memory. In *Proceedings of Designing Interactive Systems*, London, England, June 25-28, 2002, 303-311.

Gonzalez, V. and Mark, G. (2004). "Constant, constant, multi-tasking craziness": Managing multiple working spheres. In *Proceedings of CHI*, Vienna, Austria, April 24-29, 2004, 113-120.

Graham, A., Garcia-Molina, H., Paepcke, A. and Winograd, T. (2002). Time as essence for photo browsing through personal digital libraries. In *Proceedings of JCDL*, Portland, Oregon, USA, July 14-18, 2002, 326-335.

Henderson, D.A. and Card, S.K. (1986). Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics*, *5*(3), July, 1986, 211-243.

Hudson, J., Christensen, J., Kellog, W., and Erikson, T. (2002). "I'd be overwhelmed, but it's just one more thing to do:" Availability and interruption in research management. In *Proceeding of CHI*, Minneapolis, Minnesota, USA, April 20-25, 202, 97-104.

Iqbal, S.T. and Bailey, B.P. (2005). Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *Proceedings of CHI*, Portland, Oregon, USA, April 2-7, 2005, 1489-1492.

Iqbal, S.T. and Horvitz, E. (2007). Disruption and recovery of computing tasks: Field study, analysis, and directions. In *Proceedings of CHI*, San Jose, California, USA, April 28 – May 3, 2007, 677-686.

Kaptelinin, V. (2003). UMEA: Translating interaction histories into project contexts. In *Proceedings of CHI*, Ft. Lauderdale, Florida, USA, April 5-10, 2003, 353-360.

Kelly, S and Davis, J. (2003). Desktop history: Time-based interaction summaries to restore context and improve data access. In *Proceedings of INTERACT*, Zurich, Switzerland, September 1-5, 2003, 204-211.

Kidd, A. (1994). The marks are on the knowledge worker. In *Proceedings of CHI*, Boston, Massachusetts, USA, April 24-28, 1994, 186-191.

Krishnan A. and Jones, S. (2005). TimeSpace: Activity-based temporal visualization of personal information spaces. *Personal and Ubiquitous Computing*. January, 2005, *9*(1), 46-65.

Jones, W., Dumais, S. and Bruce, H. (2002). Once found, what then?: A study of "keeping" behaviors in the personal use of web information. In *Proceedings of the American Society for Information Science and Technology*. Philadelphia, PA, USA, November 18-21, 2002, 39(1), 391-402.

Jones, W., Phuwanartnurak, A.J., Gill, R. and Bruce, H. (2005). Don't take my folders away! Organizing personal information to get things done. In *Proceedings of CHI*, Portland, OR, USA, April 2-7, 2005, 1505-1508.

Landsdale, M. (1998). The psychology of personal information management. *Applied Ergonomics*, March, 1998, *19*(1), 55-66.

Landsdale, M (1990). *The role of memory in personal information management.* Academic Press Series in Computers And People, 39-49.

Landsdale, M., Simpson, M. and Stroud, T.R.M. (1987). Comparing words and icons as cue enrichers in an information retrieval task. In *Proceedings of INTERACT*. Stuttgart, Germany, September 1-4, 1987, 911-916

Malone, T.W. (1983). How do people organize their desks? Implications for the design of office information systems. *ACM Transaction on Information System*. January, 1983, *1*(1), 99-112.

Mander R., Salomon, G.. and Wong, Y.Y. (1992). A 'pile' metaphor for supporting casual organization of information. In *Proceedings of CHI*, Monterey, California, USA, May 3-7, 1992, 627-634.

Mark, G., Gonzalez, V.M. and Harris, J. (2005). No task left behind? Examining the nature of fragmented work. In *Proceedings of CHI*, Portland, Oregon, USA, April 2-7, 2005, 321-330.

Marlow, C., Naaman, M., Boyd, D. and Davis, M. (2006). HT06, tagging paper, taxonomy, flickr academic article, to read. In *Proceedings of Hypertext*, Odense, Denmark, August 22-25, 2006, 31-40.

Matthew, T., Czerwinski, M., Robertson, G. and Tan, D. (2006). Clipping lists and change borders: Improving multitasking efficiency with peripheral information design. In *Proceedings of CHI*, Montréal, Québec, Canada, April 22-27, 2006, 989-998

Monty, M. L. (1986). Temporal context and memory for notes stored in the computer. *SIGCHI Bulletin*. October, 1986, *18*(2), 50-51.

O'Conaill, B. and Frohlich, D. (1995). Timespace in the workplace: Dealing with interruptions. In *Proceedings of CHI*, Denver, Colorado, USA, May 7-11, 1995, 262-263.

Oliver, N., Smith, G., Thakkar, C. and Surendran, A.C. (2006). Swish: Semantic analysis of window tittles and switching history. In *Proceedings of Intelligent User Interface*. Sydney, Australia, January 29 – Febrary 1, 2006, 194-201

Plaisant, C., Milash, B., Rose, A., Widoff, S. and Shneiderman, B. (1996). LifeLines: Visualizing personal history. In *Proceedings of CHI*, Vancouver, British Columbia, Canada, April 13-18, 1996, 221-227.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130-137.

Ravasio, P., Schar, S.G. and Krueger, H. (2004). In pursuit of desktop evolution: User problems and practices with modern desktop system. *ACM Transaction on CHI*, June, 2004, *11*(2), 156-180.

Rekimoto, Jun. (1999). Time-Machine computing: A time-centric approach for the information environment. In *Proceedings of User Interface Software Technology*, Asheville, North Carolina, USA, November 7-10, 1999, 45-54.

Ringel, M., Cutrell, E., Dumais, S. and Horvitz E. (2003) Milestones in time: The value of landmarks in retrieving information from personal stores. In *Proceedings of INTERACT*, Zurich, Switzerland, September, 2003, 184-191

Robertson, G., Czerwinski, M., Larson, K., Robbinson, D.C., Thiel, D and Dantzich, M.V. (1998). Data mountain: Using spatial memory for document management. In P*roceedings of User Interface Software Technology*, San Francisco, California, USA, November 1-4, 1998, 153-162.

Robertson, G., Horvitz, E., Czerwinski, M., Baudish, P., Hutchings, D., Meyers, B., Robbins, D. and Smith, G. (2004). Scalable Fabric: Flexible task management. In *Proceedings of Advanced Visual Interfaces*, Gallipoli, Italy, May 25-28, 2004, 85-89.

Salton, G., Wong, A. and Yang, C. S. A vector space model for automatic indexing. *Communication of the ACM*, *18*(11), 1975, 613-620.

Sellen, A., Fogg, A., Aitken, M., Hodges, S., Rother, C. and Wood, K. (2007). Do life-logging technologies support memory for the past? An experimental study using SenseCam. In *Proceedings of CHI*, San Jose, California, USA, April 28 – May 3, 2009, 81-88.

Smith, S. and Vela, E. (2001). Environmental context-dependant memory: A review and meta-analysis. *Psychonomic Bulletin and Review*, *8*(2), 203-220.

Smith, S and Glenberg, A. (1978). Environmental context and human memory. *Memory & Cognition*, *6*(4), 342-353.

Suchman, L. (1987). *Plans and situated actions. The problem of human machine communications*. Cambridge University Press.

Tan, D.S., Stefanucci, J.K., Proffitt, D.R. and Pausch, R. (2001). The infocockpit: Providing location and place to aid human memory. In *Proceedings of Perceptive User Interfaces*, Orlando, Florida, USA, November 15-16, 2001, 1-4.

Tashman, C. (2006). WindowScape: A task oriented window manager. In *Proceedings of User Interface Software and Technology*, Montreux, Switzerland, October 15-18, 2006, 77-80.

Whittaker S. and Hirschberg, J. (2001). The character, value, and management of personal paper archive. *ACM Transactions on CHI*, *8*(2), June, 2001, 150-170.

Woodruff, A., Faulring, A., Rosenholtz, R., Morrison, J. and Pirolli, P. (2001). Using thumbnails to search the Web. In *Proceedings of CHI*, Seattle, Washington, USA, 198-205.

VITA

Name: Young Joo Park

Address: Department of Computer Science and Engineering, Texas A&M

University TAMU 3112 College Station, TX 77843-3112

Email Address: yjoo9317@cs.tamu.edu

Education: B.S., Computer Science, Yonsei University, South Korea, 2000
M.C.S., Computer Science, Texas A&M University, 2003
Ph.D., Computer Science, Texas A&M University, 2009