Title
Studying episodic access to personal digital activity : activity trails prototype

Author
Cangiano, Gaston R.

Publication Date
2011

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Studying Episodic Access to Personal Digital Activity: Activity Trails Prototype

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Cognitive Science

by

Gaston R. Cangiano

Committee in charge:

Professor James D. Hollan, Chair
Professor Barry Brown
Professor William Griswold
Professor Edwin L. Hutchins
Professor David Kirsh

2011

The dissertation of Gaston R. Cangiano is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2011

DEDICATION

To two, the loneliest number since the number one.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thanks to whoever deserves credit for Blacks Beach, Porters Pub, and every coffee shop in San Diego.

Thanks also to hottubs.

ABSTRACT OF THE DISSERTATION

Studying Episodic Access to Personal Digital Activity: Activity Trails Prototype

by

Gaston R. Cangiano

Doctor of Philosophy in Cognitive Science

University of California, San Diego, 2011

Professor James D. Hollan, Chair

It was just a generation ago that computers entered the workplace. Back then, they only *represented* the work we did, nothing else. But today, some sort of computing device is involved in how we play, how we communicate, how we get our news and of course, how we work. What this means is that today almost all aspects of our lives are *represented* in some digital form. The rapid pace of change in technology and the dramatic shift in the use of computers has a cost associated with it. The legacy design of early computer systems is still prevalent in modern devices and goes unnoticed because of our familiarity with it. The desktop metaphor with its file and folder system, and the application paradigm with its document-centric view of information, both carry the legacy of a design

that has far surpassed what it was originally designed to do. Digital representations should mediate what we do in the physical world, and since we do much more now than just work through computers, we need new representations that leverage our cognitive abilities in everyday life; in particular, present day computing devices do not facilitate the use of a powerful skill we use in our personal experiences, known as episodic memory. Episodic memory is how we remember our lives through stories.

The human perceptual system samples the world continuously in order for the brain to store information, organize it and later recall it efficiently. At least this is the classic view of memory. However, people also leave a physical trace behind each and every one of their actions simply as the byproduct of their interaction with the environment. Because memory is finely tuned to reconstruct the past, our perceptual skills help us make meaning out of these traces. Time, proximity and familiar surroundings provide cues that naturally trigger our recollection of the past. Episodic memory is a human skill that taps into these cues by encoding the context surrounding events therefore allowing us to re-experience the past by recalling specific instances and the context in which they were experienced. Computers, in contrast to humans, only record the *consequences* of our actions and in doing so, they reduce the type and quantity of the memory cues available.

Things have indeed changed dramatically in the last generation. Together with the avalanche of new digital gadgets and the general overwhelming presence of computers in our lives come new opportunities for research. One new and exciting ground for research is the use of activity recordings. The main research question of my work is understanding how to design an interface using these recordings to facilitate the use of episodic memory. There is a growing interest in the field of Human-Computer Interaction in research of life-logging technologies to assist with memory and reflection. But we are only at the very onset of understanding the impact of these technologies, and more importantly, how they might fit into the fabric of our daily lives.

The work I present here was motivated by a year-long ethnographic study conducted at a law office. In this study, I used desktop activity recordings as a novel methodology to learn about the nature and details of work. I learned that what is usually considered *multitasking* behavior in the literature, is in fact the norm in this setting. Multitasking here is not "crunch mode" type of behavior, but is a self-selected and all together different kind of work style. This style is engendered by both the nature of the legal work and the new digital tools available, in particular communication tools such as instant messaging and email. These tools have had an impact in how paralegals and attorneys interact with clients and with one another. My ethnographic data reveals that with the increasing frequencies and flexibility of the daily interactions comes an increased *fragmentation* of the context of each work thread. The lack of episodic support in these tools creates a heavy load for workers. Paralegals and attorneys have to put effort to bring together the *history* of a case from the many separate pieces of the past (email, instant messages, database entries, and so on). In other words, workers have to *build context* for a case before communicating with clients and this context consists of putting together a *timeline* representation about the history for a case, containing a chronology of events with the client and a view of any upcoming deadlines or pending issues for the case. My argument is that the tools available presently do not support this context-building process, so in addition to supporting for multitasking and interruptions we need to design support for this process.

The second portion of the work in this thesis brings my findings to bear directly on a software design problem for Human-Computer Interaction. It describes the design and implementation of a software prototype tool called *Activity Trails* with the goal of supporting episodic memory. The thesis ends with a study conducted with researchers at UCSD evaluating the benefits of episodic access for everyday activity through detailed case studies of usage.

1 Introduction

The interface design for most digital devices — and in particular the prevalent application/desktop metaphor — supports mainly the use of semantic memory. Every time we access our lives through these interfaces, we are confronted with a design that relies on the use of semantic categories and labels to organize our information. Semantic knowledge is what we *know* about ourselves (our organization) and the world. The folder structure in our personal computers for instance, reflects how we categorize our files, and each application we use has its own set of semantics that reflect common notions of what we have come to learn through years of computing use. We are not aware of this because we have grown accustomed to this legacy design in computers, but there is more than semantic knowledge involved in how we organize our lives in our minds. There is an important cognitive difference between what we know and what we actually *recall* from our daily experience, as well as that which we can infer on demand by associating whatever cues are available to us from our recollection. Traditionally in computers we save an important document in a folder that bears the name of the project we are working on, or we bookmark useful web pages in our browsers. Now with the advent of the Web 2.0 it might be easier to just Google a name or use *tags* to find the things we need when we need them in a variety of socially maintained content (such as Delicious, StumbleUpon, and so on). But even for this loosely defined set of " tags", the organization of our *stuff* needs to follow some general schema, either defined by us beforehand or as shared knowledge with the rest of the world. But more importantly we need to *explicitly* tag what we use. This is not the case for a lot of what we do in the physical world. We are not necessarily aware of everything we will remember, and most importantly don't have to make a conscious effort to tag something in order to remember it. Our memory has evolved such that what we forget is just as important as what we remember, given that we only have a limited capacity to readily remember names, dates or words. But it is remarkable how much we can actually remember in our lives. Episodic memory is a human skill that allow us to tap into our past using a very different organization than semantic categories and labels. Episodic knowledge contains information about the surrounding context of instances we experienced. It contains information in the form of perceptual cues, such as vivid images, and these cues are part of our daily surroundings. As we go about our business, we leave behind traces as we interact with the environment and these traces are cues into our episodic mental representations of the past. Computers presently provide poor support for episodic cues, putting the emphasis on labels and categories.

Part of the lack of episodic support in current computer design, is inherent from its *application/document* metaphor which establishes a set of artificial boundaries that fragment episodic recollection. Events that are meaningful to us, such as making plans or researching a new investment for example, are distributed across a variety of digital resources like emails, instant messages, browser bookmarks, websites and so on. In order to trace back to these events, we are required to recall both the general procedures we used to generate them (the applications) in addition to the item-specific labels we assigned to each of their parts, provided we explicitly saved, tagged or bookmark them. In addition, we need to put all these pieces together in order to get a full picture of the recollection we have of that event.

In order to design effective support for episodic memory, graphical interfaces need to provide access to the *histories* of our daily activity. One of the most surprising discoveries from my field work was that when people saw recordings of their own desktop activities, they often experienced *vivid recollections* that extended well beyond what was on the computer screen. For instance, people recalled phone conversations, packages arriving in the mail, and face to face conversations that preceded or followed the images on the screen. The potential benefits of this type of episodic access are exciting: to be able to explore our own past and to search for things we cannot recall well but that we can identify once we see them; to reinstate the full context of past activity in order to better continue in the present; or to benefit from previous efforts by applying them to our current work. But this is much more than just creating a time-machine. First of all, in order to effectively use the past, we need to have more than just access to specific dates and times. Most of the time we actually do not recall that type of information very well. What we do recall are instances of activity around the dates and times in question. Playing back the past in real-time is too slow a process and too overwhelming to be effective if we do not know where to look. What we need is the ability to quickly browse through brief episodes which contain similar content around a loosely define time frame. Perception and memory do the rest of the work through recollection by association and context. The approach I am taking in this proposal work is to provide summarized instances of recorded activity on computer desktops. These summarized *episodes* expose chronological ordering to the user and allow them to quickly browse past activity by zooming in and zooming out of ordered sequences of short visual episodes. In addition, the graphical user interface of the Activity Trails prototype allows users to search through time using keywords and familiar terms.

1.1 Work Scenario

The following scenarios stem from interviews conducted at the law office. The first one illustrates the current situation workers face when collecting background on a case, and the difficulties they encounter with the tools they currently use. The second scenario describes an ideal scenario using Activity Trails. The contrast of the two highlights the lack for episodic support in the digital office tools available today.

1.1.1 Current Practices

RR is a senior paralegal at the Law Firm. His primary job responsibility is compiling the material required to file client cases. This includes government forms, work and personal documentation, drafting letters of intent and other written material. His daily work consists of following up on the status of cases, including email correspondence and phone calls, answering clients' questions, notifying clients about case progress, keeping track of filing deadlines, and scheduling meetings or conference calls. Most of this information is kept in a client database where he annotates each interaction with a client by either cutting and pasting from an email or writing down a summary of a phone call. But in addition to this database, he keeps a well organized set of email folders categorized by company name, which he often navigates or searches in order to find previous correspondence with clients. He also maintains a case log in an Excel spreadsheet which allows him to monitor progress at-a-glance on current cases (this is a common practice in this office). In order to ensure uninterrupted service to clients, paralegals are required to share cases, so these case logs are stored in a common internal server accessible to everyone. In addition to and because of this sharing practice, RR and his colleagues are constantly instant messaging each other, to request information about cases. Instant messaging is therefore an integral part of their work flow and constitutes a permanent record of the background information about cases, albeit one that is hard to store and organize, so it is often lost.

A major portion of RR's daily activity is emailing with clients. Therefore he spends a significant portion of his time going through his inbox, replying to inquiries and initiating deadline and update notifications. The following excerpt describes such a situation

> *"...so here I was supposed to follow up on one of XXXX's (paralegal's name) tickle hmmmm ... on whether or not this guy wanted to start a merit-based greencard so I am trying to look for the old email ...so I had to go into my email to pull up a summary of the previous correspondences but if it's older than six months ... I usually delete emails older than six months because our server slows down"*

The screen recordings show RR navigating an elaborate hierarchy of email folders in MS Outlook, briefly opening emails to browse their content, until he finds the ones he was looking for. Very often RR also asks colleagues directly for background information on a case, whether because he does not remember, cannot find it, or because the case is shared. In these cases RR work is interleaved with IM conversations, and this messages constitute tacit knowledge distributed within this organization that is not otherwise recorded in any of the digital tools available (database, emails, case logs, etc). There is also a client database, which contains the background information for a case, but not in a format that supports a clear view of the *history* that case, as explained in the following excerpt

> *"... looks like something was lost ... and so we are trying to figure out ... not lost necessarily but trying to figure out where the package was ... sooooo I am looking into her case history (database) aaaaaand trying to ... (long pause) trying to refresh my memory about her case before I responded to her... the one database that we have ahhh online database does not allow one window (pause) more than one window to be open so we have one part of the database has a log of the conversations we had all the emails exchanges strategy notes and then we have the outlook calendar for tasks list for update for thing we have to do next we have a case log but the the online database ahmmm (pause) it's it takes too long and you constantly have to (inaudible, while he motion with a "check" gesture) two different fields you have to type the first name second name or pieces of it to get that person up and then once that screen is up there are many different chambers (waving hands) within that screen to just to find out what had occurred let's say a month ago so if there was ISSUE in the case, it's buried in that database..."*

Paralegals at this law office have to juggle between several different applications and tools to be able to put together the history of a case. As the previous excerpt illustrates, the "ISSUE" about a case is not something that could easily be retrieved from a database, but rather, the information needed to put that case into context (and therefore retrieve the issue in question) is fragmented across database fields, email folders and instant message conversations. This is a type of scenario that workers at this office deal with often.

1.1.2 Activity Trails

RR is again responding to client emails. It is morning time, so his inbox is full and there is plenty of client correspondence to be addressed. He quickly opens each email, briefly responds when appropriate, or marks them for reply later. While going through his inbox, RR often opens instant messaging windows and talks to his colleagues in the office. When he reaches a certain email, he pauses to read it carefully. A client is inquiring about a job order letter (a letter drafted by the law firm as required by the government to serve as a public record of the job description and availability). As it turns out, this letter was slightly delayed, so the client is expressing concern. After reading the email, RR tries to recall the details of this case, but he has not touched it in over a month, so his memory is not fresh. Instead of looking for previous emails in his folders, RR clicks on the Activity Trails icon on the lower right-hand corner of this computer screen to activate it. He continues to type the name of this client and hits the `ENTER` key. The Activity Trails program searches through his activity histories, and after a few seconds returns by displaying a sequence of thumbnails on the lower portion of the screen as seen in Figure 1.1. Each thumbnail represents an episode of activity, in the form of a summary movie clip of the desktop. RR rolls over these thumbnails and sees a small preview of each episode. Each picture indicates a date, so that RR gets an idea of when that occurred and the view of the overall sequence gives him an idea of the development of that case during any given day. He can see himself browsing a company's website, as well as a government site with information about employment categories. RR spots a thumbnail where he can see that he was writing an email to this client, roughly a month ago. He double clicks this image and the image expand to full screen size and playbacks a few minutes of activity slightly accelerated. He can see his own drafting of the email, and at some point the writing stops. He thinks to himself

> *"...so that's what I was typing there (pointing to the screen), and I got off the phone with I remember getting off the phone with her and I remember trying to do the job order and they calling me and me calling her back and we" re trying to get this letter to her and so we typed it and sent it ...I think what happened was I called her to say I want the job order to be for less time because they had a long period of time and when I did that she says 'who are you?' and i say 'oh, I'm from the law firm' and she says 'ok then we need a letter from you before we talk to you' so I drafted the letter and sent it over to XXXX (company's name)"*

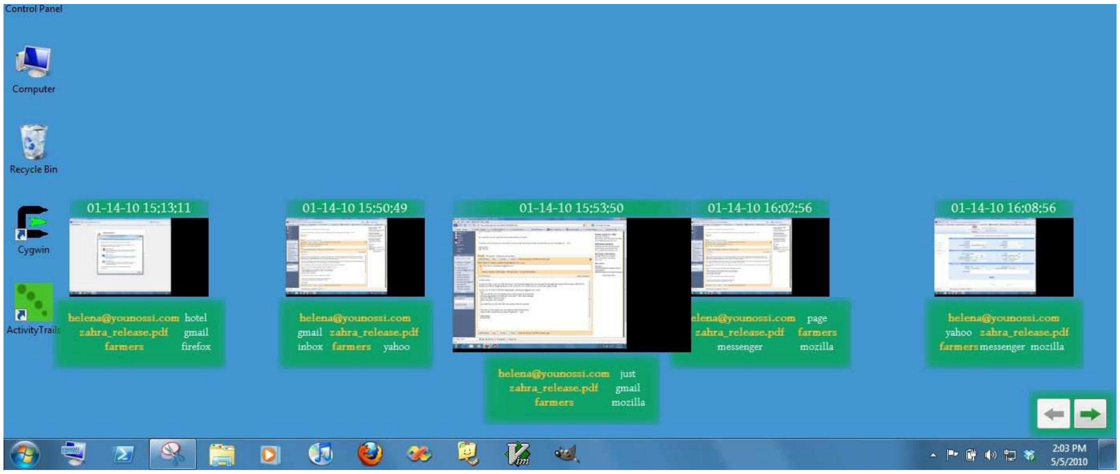

Figure 1.1: Graphical User Interface for Activity Trails. Thumbnails float on top of desktop representing a visual chronology of evens. Users can scrub thumbnails to get a preview of the activity or double click them to see a full screen summary playback. Each thumbnails also display the time and date of the event, as well as the most frequently appearing keywords from the user's typing or reading.

RR now clearly recalls the incident that led to the delay with this letter. He understands the potential sensitivity of the client towards the delay, and that he should mention something about expediting communication between the two offices for the future. RR now closes the full screen preview and goes back to the thumbnail view on the lower part of this desktop, and clicks on the next thumbnail to the right. He again sees himself writing an email, but this time he spots the word "paystub" which triggers his recollection of another related incident. He recalls the following

> "...so this guy calls or I called him so then the form or the forms came in that Fedex that we were waiting so then I called him to get a paystub so we can file the case today and then he is saying about the XXX (company name) bought out his company XXX (other company name) or whatever...'cuz I just saw the word 'paystub' and then i remember that this guy works for XXX (company name) ...(long silence pause) I am on the phone with him I'm on the phone with him and IM'ing XXX (attorney's name) and then because he wanted me to (pause) i told him to fax it to me and he said no and i just wanted to file it so it was like i'll just do the research versus waiting for him..."

It is now much clearer to RR what the story with this client was. In this instance, he did his own research rather than wait for the information to arrive on the mail. He thinks for a few more minutes and is now ready to answer this morning's email.

1.2 Thesis Statement

This work is based on the following premise:
"*Software support should strategically target the unique characteristics of human memory and to be concerned primarily with designing effective retrieval cues rather than with "capturing all experience as a form of memory aid.*"[69]

My claim is that the lack of episodic support in the design of personal computing devices is a valid research problem, so I propose to study its role in the fabric of everyday digital activity and how to support it through the use of software. There are two assumptions underlying this claim, which this work will show to be true. The first one is that computing devices don't directly support the use of episodic memory to access our personal information. The second assumption is that direct support for episodic memory benefits users within the existing ecology of applications and activities. I will be presenting field work that highlights specific areas of current practices at a law office that could greatly benefit from episodic access and how the current digital tools available don't support the episodic aspects of the workers' job.

1.3 Research Approach and Contributions

Together with the increasing number of workplace studies, there is convergence between academia and the industry toward a common analytic agenda to " respecify" technology with regard to human practice and social organization [53]. The social sciences on the one hand have moved toward a situated explanation of the contingent nature of commonplace work behavior through detailed observational studies. They have also shown a move to study the mediating role that artifacts play at the center of analysis, therefore shifting the study of technology as constituted in social interaction [36]. Industry work on the other hand has shown a move toward the social, not only by interest in naturalistic observation of workplace activity, but also driven by a need to develop less stipulative analytic methods for the design of technology, which where originally established by Cognitive Psychology. The rapid changing face of technological development, and the large diversity of workplace settings and work practices make for a very fertile and exciting ground for research.

One contribution of my research is to bring observational data to bear directly on a Human-Computer Interaction design problem. I consider the law office setting as a modern high-paced work environment, where digital technologies are employed to solve real-life problems in ways that engender new work patterns. The traditional analytical model of work activity — a person switching from task to task sequentially driven by interruptions — is not the most appropriate description for a setting like this one. Multitasking is the *norm* and not an exception that occurs only during crunch-time. To provide software support for this type of work behavior requires us to reconsider the nature of our behavioral models in HCI. Instead of designing to support task resumption, we demonstrate that context can also be supported by considering it as the rich history of interaction with a particular entity (in this case, a foreign national or company client).

Another contribution of this work is to the HCI community with the design and testing of a novel interface to personal digital information. The need for episodic access to information has been previously noted as a challenge area in HCI [68][10]. Only a limited number of interface designs and evaluations have been conducted on this topic, so this work makes its contribution with a design grounded in observational data and examples of real-life limitations where episodic access in personal computers can be an important benefit for users. It also contributes with better understanding of the design requirements for building modern episodic interfaces and how these might fit into the fabric of everyday work, together with the existing ecology of tools available today.

1.4 Thesis Organization

Chapter 2 presents the theoretical background needed to understand the cognitive basis for the link between episodic memory and context reinstatement as a phenomenon observed in our data. This chapter also presents previous work in HCI that relates to this thesis. Chapter 3 presents the field work conducted at the law office, including a scenarios of the different work roles, and excerpts from interviews. The examples from the contextual interviews serve to illustrate specific difficulties that workers have using the available tools to put together the context for the history of a case. Chapter 4 describes the details of the implementation of Activity Trails, including its data structures and the computational algorithm for parsing data and creating summaries. The final chapter of this document presents the evaluation study conducted with Activity Trails, describing the areas where the tool enhances the use of episodic memory for the task at hand, and areas for improvement.

2 Background

2.1 Analytical Models of Behavior in HCI

In a broad field like HCI there are surprisingly few analytical models of behavior to help us understand the complex fabric and situated dynamics of real everyday work. This affects how we design for support. But we cannot attribute this scarcity of models to a lack of field studies, since an increasing number of observational and ethnographic work has been conducted in the last twenty years, involving the range from aviation and control rooms, to a variety of office work. Nonetheless, there are still some commonplace assumptions in HCI that hinder the community from developing a comprehensive analytical framework.

One of these assumptions stems from the prevalent model of *task / interruption / switch* behavior. This model carries a legacy from the Cognitive Psychology origins of HCI, and it presupposes an ideal situation where individuals work on single well-defined "tasks", one at a time and in sequential order. Interruptions in this model are deemed disruptive and counter-productive, something that adds to the cognitive load of workers since they have to switch the context of their ongoing work. "Multi-tasking" in this model, is an *accelerated* version of sequential task behavior.

Even though this behavioral model is appropriate for certain types of settings, work types and people's preferences, it is by no means all-encompassing of an evolving world. With the rapid development of technology and its introduction into the workplace, the landscape of work is changing even faster. Many modern offices exhibit a very fluid and collaborative type of behavior, where interruptions are often self-initiated, and constitute the very fabric of how people work by facilitating collaboration and sharing of information. In addition, the ease of access to information provided by web-based tools, encourages the management of multiple concurrent activities, something that is usually put under the umbrella of "multi-tasking". In this thesis I question the notion of multi-tasking and argue that it might be better to consider it as an entirely *different* kind of behavior, with an additional set of needs for design of support.

In my field data, I found out that what might appear to be "multi-tasking" behavior at first, is really the common behavior, and not *perceived* by workers as multi-tasking. The billing structure of this type of law, together with the highly collaborative nature of the work practice and the new digital tools available engender a type of behavior where very frequent and short interactions with clients and coworkers is the common scenario. These interactions occur in an interleaved fashion throughout the day. The *task / interruption / switch* model presupposes that at any given moment a person is working on a single task, statically defined by a specific set of documents, applications and people, and that an explicit switch must take place for a new task to replace the old one.

Delineating what a "task" is at any given time is very difficult, even for workers, since it involves interaction with *overlapping* sets of multiple people, applications and documents. Even if a task could be delineated in a scenario like this, providing support for switching contexts rapidly might not be the only kind of support we can provide. Workers are constantly switching from one resource to the next one in order address the momentary needs of the task-at-hand. Another way to support this kind of behavior is to support *weaving* together the fragmented context of activity emerging from the use of the tools available to get the job done.

2.1.1 Multi-tasking and Interruptions

Perhaps one reason for the overuse of the *task / interruption / switch* model in HCI are misconceptions about the notion of multi-tasking. Usually multi-tasking is considered to be the observable behavior of a person who appears to be handling multiple activities or tasks simultaneously. This is reflected in the definition from Wikipedia: "*the performance by an individual of appearing to handle more than* *one task at the same time*" (note the use of the word "appearing"). But humans have only a limited set of attention and memory resources to work with, as many years of research in Cognitive Psychology have shown. The attention required to process a task is constrained by a "bottleneck" to sequential processing [9][80]. The type of memory that is readily accessible for the task-at-hand (known as working memory) is limited to a few items of chunks of information at any time [55][1]. But skilled workers appear to handle multiple tasks seamlessly, and this is specially true in modern settings that rely heavily on new web communication tools. More generally, an entire new generation dubbed "generation M" [38] or the *natural multi-taskers*, seem to effortlessly thrive on this type of behavior. So why then do HCI researchers still use a behavioral model that hinges so heavily on this questionable definition of multi-tasking?

One answer to this question could be in the remnants from the origins of HCI. The Model Human Processor [14] was the predominant model for studying behavior during the times when Cognitive Psychology gave HCI its original impetus. But in this model, multi-tasking was perceived as an accelerated case of normal sequential task processing. That is, since humans cannot process more than one task at a time, they *alternate* their attention in a round-robin fashion, much like computer processors do, borrowing a metaphor from Computer Science. That is, a person works on a single task at a time, and maintains in working memory a set of related goals, strategies and bits of information that constitute the context of the work-at-hand. At any point, an interruption causes the person to switch to a different task, incurring in the cognitive load of having to change the contents of working memory, and additionally shift his attentional focus. This process is therefore costly if interruptions are too frequent, unlike CPU's. This type of Computer Science metaphor seems a bit outdated in today Cognitive Psychology's terms. Its legacy nonetheless is still felt when researchers in the HCI community devote so much effort to designing software to support multi-tasking and interruptions, with the emphasis on aiding workers "recover" quickly from the perils of context switching. This is not to say that there is no cognitive cost associated with handling multiple tasks in a seeming concurrent fashion, but perhaps there are other ways to support this type of behavior if we use revised models more suited for newer work settings and practices.

Another way to consider what appears to be multi-tasking is as a different type of behavior altogether. Perhaps is a kind of behavior inherent to some category of people, a cultural preference, and/or a work place or technology-driven self-selected behavior. This notion is not new. Strauss was one of the first researchers to observe that work can be seen as *trajectories* of multiple tasks [73], and since then, several workplace studies have shown that office professionals, and in particular managers, unknowingly benefit from multi-tasking and from the flexible work flow driven by spontaneous interruptions in the workplace [72][60][65]. This could be a kind of behavior more suitable for activities that require building human relationships as well as solving complex problems, as managers do. In these studies participants often report that interruptions are actually the "fun stuff" of work, or that they produce overall positive results by helping teams synchronize their efforts. More generally, ethnographic work has started to revealed that this could be in fact part of the nature of how people do things. The terms *polychronic* and *monochronic* have been used to described tendencies in the natural flow of work across cultures, workplaces and most interestingly with the introduction of new technologies [46][34][49].

One relevant example is work done by Barley with workers at two community hospital radiology units in Massachusetts [4]. In this study Barley compared a traditional X-Ray unit with new CT scanning technology. In the traditional method unit technicians were often consulting with radiologists, so their work structure is fragmented by frequent interruptions of relatively short duration and unpredictable sequence. Technician's work by contrast, is highly structured by the patient's appointment schedule as well as the X-Ray procedures. Radiologist therefore have a self-selected polychronic behavior, and thrive in a flexible yet unpredictable work temporal structure. Technician's time by contrast is highly constrained by the sequential nature of appointments and the rigid structure of the radiology procedures. This contrast creates an asymmetry between the two groups of workers which results in tension and conflict between them. Technicians are often frustrated and unhappy with their work relation because they have to search around for radiologists on duty who might be on the phone, or consulting with doctors, or reading film. With the introduction of CT scanning and ultrasound technology, radiologists' temporal structure of work has changed. Task duration has been lengthened and thus task switching slowed down. Radiologists now mostly consult with physicians only, and follow the patient appointments schedule more closely. The introduction of new technology in this setting has changed the work behavior of radiologist, making it more monochronic and thus more in tune with the work pattern of technicians, with overall positive results.

If we are to consider polychronic work behavior as a distinct phenomenon, then we need to also consider the kinds of work settings that engender this behavior, and also importantly, the role that technology might play. The data presented in the next chapter shows how internet-based communication tools have changed the way paralegals and attorneys work. The law office under study exhibits self-selected polychronic work habits because the new digital tools employed there support it. Recently other researchers in HCI have also began to incorporate these new notions of behavior into their analysis. This is a move away from older stipulative models trying to define what constitutes a "task", and a move closer to developing more situated analytical models. One good example is the work by Mark and colleagues [33][54]. Their field work highlights how information workers maintain several streams of concurrent activities throughout the day, driven by unexpected and self-initiated interruptions. They introduce a new concept they call "working spheres", as a unit of analysis based on a sphere of collaborations, resources and time, which workers have to manage throughout the day. A working sphere replaces the notion of a task with a conceptual unit of work that is situated by including time and interaction with people. Mark et al. [74] point out how communication between workers plays an important role in how they manage multiple projects. Communication provides *alignment* with others involved in a given sphere in order to help maintain an overview of progress as well as to resume work more easily. Previous ethnographic work underlines also the role of informal communication in problem solving and flexible adjustment to changing circumstances [48][81].

Communication plays an important role in polychronic behavior, specially in highly collaborative settings. In addition, the role of new technology — notably email and instant messaging — has engendered new work patterns that support polychronic behavior. These have yet to be incorporated into new analytical models for HCI. In our field site for instance, the introduction of email has encouraged much more frequent communication patterns with clients than was possible in the past with the use of the phone only. Email has significantly changed the client-attorney relationship in the last few years. Attorneys can now delegate client communication to paralegals since it is easier for them to review and standardize email language that go out to clients. As a consequence, paralegals' work has also changed from processing documents and data entry to answering and drafting emails for a big portion of their work days. This shift encourages paralegals to adopt a more polychronic behavior, moving from sequential processing of data entry jobs to handling incoming emails form clients in a manner that is less predictable and requires more frequent and short interactions. Instant messaging in addition has also changed the pattern of work at this office. Both paralegals and attorneys have been particularly keen in adopting instant messaging as part of their work flow. Our data shows paralegals frequently messaging coworkers regarding cases, even when sitting in physical proximity of each other. They explain that an instant message is a lot less intrusive than walking into someone else's office, and so encourages brief and frequent informal exchanges. The structure of this office is such that paralegals are expected to share cases as much as possible. It is a lot easier now for a paralegal to quickly check on a case via instant messages with others involved. Instant messaging therefore encourages a behavior that might seem at first hand more disruptive, since message windows appear frequently on people's screens. But these seeming interruptions have now become an integral part of the work flow, are mostly self-initiated, and contribute to a well coordinated office where cases can more easily be shared among workers. In sum, this office has acquired a self-selected polychronic work pattern partly due to the nature of the legal work, and partly due to the introduction of new technologies.

For our analysis, we consider polychronic behavior as concurrent task *man-*

agement, in contrast to concurrent task *performance* [52]. In their study of aviation pilots Loukopoulos et al. (idib.), show how highly efficient multi-tasking pilots are effectively managing several goals, time frames and resources through planning, strategizing and prioritization of actions while flying. Similarly, we will present verbal data from a law office describing how workers manage multiple activities simultaneously. In order to make best use of their time, paralegals and attorneys have to manage the heavy load of cases and the unpredictable temporal structure and ordering of incoming emails, alerts, interruptions and phone calls.

2.2 Related Work in HCI

2.2.1 Task Support Systems

Research on task support has increased over the last five years in HCI. The idea is not new and dates back to the seminal work on the Rooms Workspace Manager [15], but there is renewed interest in this area due to growing empirical data on the nature of modern office work [54][22][42] and an increasing concern about the cognitive costs of multitasking [2]. As mentioned in the previous section, the research for supporting multitasking is dominated by a behavioral model based on tasks and interruptions, so most of the work produced in this area is in assisting task switching and restoring associated contexts. In this sense, my work is a contribution to the field of HCI with a complementary model to help better understand and support real-life work.

Task support research is based on the notion that a "task" and its associated context can be represented as a "state", composed of a given number of resources, such as windows, documents, browser tabs and so on. One of the most common approaches therefore has been to support grouping windows [59][26][63][76][3][62]. For example, the SWISH project at Microsoft [59] applies machine learning technology to cluster windows based on their title and temporal usage proximity. Once these window clusters are determined, users can save and restore window configurations, in order to switch back into a previous task. Other approaches extend the same notion by providing users with ways to organize unstructured data beyond documents and windows, such as communications, without restrictions on the type of resource involved (e.g., email, documents, IM, web pages, etc)[56][45][24][27]. For instance, the Haystack project [45] provides a sophisticated user interface that can access almost any element from users' activities. Haystack provides the facilities to view emails, chats, web pages and documents under user-defined categories. In principle, this type of approach solves many of the support challenges for modern work, but there is an important caveat: users are required to explicitly define the parameters of the associations between resources (i.e., the categories). Having to manually define every context, by tagging or annotating it, significantly increases the cost-benefit ratio of the approach and the likelihood of adoption.

One area that is the more closely aligned with the approach of this thesis is the Vibe Group at Microsoft Research (Visualization and Interaction for Business and Entertainment). VIBE focuses on the visualization of a wide range of areas in Human-Computer Interaction. In particular, a project named Personal Vibe is of direct relevance to my work. Personal Vibe [40] is a single user, single machine personal database of activity. It records windowing events on a continuous basis and provides a set of functions that developers can use to obtain information about user patterns. One of its application is a tool called Status Writer [11] that provides support for users to reflect on their time spent at work. This tool shares many similarities with Activity Trails. It displays thumbnails with screen contents at various moments in time, and can parse activity in terms of applications, documents, web pages and so on. Status Writer also has the ability to playback entire segments of recorded desktop activity (in very low resolution). The main differences between Activity Trails and Status Writer are in the ultimate goals the two tools are designed to support. In the case of Status Writer, as the name implies, the main goal is to support reflection about activity in order to write summaries or time sheets, so the tool has charts visualizing distribution of time and an interesting visual representation for frequency of usage (a pile metaphor). Activity Trails on the other hand is designed to support exploring, searching and reinstating the past. It provides high resolution full screen playbacks and the ability to easily jump across time and view histories of episodes in the past relating to same items.

Microsoft Research also has a long history of related research. Dumais and Cutrell [27] developed a system called "Stuff I've Seen Before"(SIS). The original impetus of this research was to blur the boundaries of applications by enabling searching on a system-wide level and based on familiar terms to the user. More specifically, SIS let users search on whatever they remember about any item they need to retrieve, whether it be an email, or a web page or a local document. In its second incarnation called PHLAT [20] Dumais and Cutrell continued with the same idea but adding powerful tools like tags and filters. As stated by them, PHLAT attempts to *"blur the distinction between searching and browsing, and making it expedient to summon disparate types of relevant information by means of a simple, intuitive tagging system"*. Both SIS and PHLAT share this common view with Activity Trails that searching and browsing should be considered part of the same activity, and that most of what we do spans across applications so that it makes more intuitive sense to let users search transparently desktop-wide. The latest tool in this lineage is called "Personal Narratives"[47], and it takes these ideas a step further by attempting to create a user model based on word frequencies. The tool provides visualization of topics over time, using an desktop-wide index of 200 random words extracted from every document, email or web page indexed by the Windows Desktop Search index. Using a Windows Sidebar Widget that floats on the desktop, the tool compares terms being typed by the user with a word frequency model that represents the most commonly used terms. By comparing words against the model, the program can obtain measures of saliency to detect new terms as they appear and display those to the user unobtrusively. User can then explore the history of those terms in a visualization interface. This tool shares common ground with Activity Trails because it allows to see histories of keywords. Activity Trails attempts to not only show the histories of terms, but their usage in the context of the desktop throughout time.

A group of researchers at Google have investigated a similar concept to Activity Trails specifically focused on the activity of early web research within the browser environment[50]. The Research Trails prototype creates visual summaries of web history without requiring users to explicitly define beforehand the context of their activity. In this sense this program shares some fundamental similarities with Activity Trails in that trails of activity are produced without the user's intervention. Their research also shares the fundamental premise of my work that there is a definite need to support the process of resuming a previous task, using visual images to promote context reinstatement. The algorithm behind Research Trails automatically segments trails based on time, and then clusters the results using a version of latent semantic analysis on the contents of the web pages. Of course, one important difference with Activity Trails is that this research pertains only activity within the web browser, whereas my research is a desktop-wide analysis. More importantly though, the version of Activity Trails for the Android Mobile Phone will extend the reach of this work beyond the computer into the realm of physical context and daily general usage.

2.3 Human Perception and Memory

2.3.1 Episodic Memory and Context Reinstatement

The literature on autobiographical memory indicates that people are good at remembering the histories of events in their lives [64][6]. At the same time, we seem to have fairly poor memory for the exact dates and the locations of those events, except when those have intrinsic perceptual or emotional saliency, known as "landmarks". Autobiographical memory is organized around personal themes based on the structure of everyday life experience. Experimental work has shown how that the classic exponential memory decay curve becomes almost linear when people are cued with items like personal names rather than locations or dates [7]. These effects are robust, and produce consistent biases when people try to estimate the time of past events [41]. Similarly, because a variety of actions often occur in the same locations, a frequent location will be a poor predictor of a particular memory, but a particular memory might be a good trigger for a particular location [8]. In general, infrequent thoughts or events are good memory triggers against the background of the regular structure of our daily routines. Common daily events such as the "work review meetings on Wednesdays", or "Sunday bike rides", serve as entry points into our personal narratives and allow us to recall more details about the past.

Within the autobiographical memory system, episodic memory plays an important role encoding information about discrete events. Episodic memory is a unique human skill that allows us to "time travel" and to re-experience the past [77]. One of the prominent features of episodic memory is that it encodes information about the context surrounding events, including spatial information [57]. Access to our past is not uniformly distributed. Empirical evidence shows that access to our memory is biased around "landmarks" and that using public events or unique personal experiences increases the accuracy and content of long-term and episodic memory [70][51]. This means that people are not only better at recalling the past around salient events but that they will also remember the context surrounding those events. The main theories of autobiographical memory consider the process of reconstructing the past an integral part of how we remember [8][6] [19]. In particular, as time goes by, we rely more heavily on the organization of our memory to put back together a recollection that contains the characteristics of the original event. Memory landmarks and the ecology of our environment play a pivotal role in this reconstruction process by providing entry points into our mental representations of the past. Through the use of episodic memory, landmarks, and mental imagery, we are able to experience vivid recollection as if "being there" in the past. Episodic memory provides the pathways to and from our memories into the narratives about how we relive the past. In this sense memories are not items we retrieve, but rather the mental products of how we recall [35].

Event segmentation Theory (EST) argues that the human perceptual system spontaneously segments activity from ongoing processes regulating attention and memory [78]. Activity boundaries or "breakpoints" arise due to low-level perceptual cues consisting of abrupt changes in moment-to-moment levels of activity. These breakpoints are akin to contours in object recognition (used in computer vision models), but the dimensionality is defined by motion, frequency and object-scene composition. There is set of studies in social psychology demonstrating consistency in the way people segment sequences of social interaction episodes. For example, in an experiment conducted by Newtson and Engquist participants manually segmented short movies in which actors performed everyday activities [58]. Participants were prompted to segment the films at points they considered to be natural breakpoints. The breakpoints were highly correlated across participants, and furthermore, memory recognition increased for segments around breakpoints. Similarly, experimental work on film summarization has explored the relationship between these naturally occurring breakpoints and memory for events. For instance, commercials inserted at breakpoints are remembered better than the same commercials inserted at non-breakpoints [32]. Effective visual summaries can be created by applying relevance-based selection of salient moments of activity based on natural breakpoints. Schwan and Garsoffky argued that keeping the breakpoint structure in summaries maintains the temporal distance and perceptual cue distribution necessary to facilitate the reconstruction of past experience [67]. These findings are in agreement with the notion in EST about natural perceptual units of behavior and their relationship to memory. From the theoretical perspective of EST therefore we can think of *reinstating context* really as an act of "re-perceiving". In other words, perceiving again the same structural and temporal relations used to guide attention and memory during the original performance of the activity [82].

2.3.2 Episodic Memory and Activity Logging

There is a long history and recent resurgence of interest in recording personal data since the days of Vannevar Bush's Memex [12]. In HCI, the idea of using logging technologies to enhance memory dates back to work done by Lamming at Xerox EuroParc [10] and more current work includes the MyLifeBits project [29] and the recent series of ACM CARPE workshops [30] on capturing, archiving, and retrieval of personal experiences.

Lamming and colleagues(ibid.) set up a series of studies using logging devices to study the ways in which prospective and retrospective memory could be supported using a variety of logging devices, such as notes, images, and sound recordings. In these studies participants were able to take notes and set reminders to themselves as well as capture images and sounds in the environment using a range of technologies from smart badges to hand-held digital diaries. These devices, such as the Xerox Tab, were capable of sensing and react to the surrounding environment creating context-aware reminders that provided contextual triggers for memory. Lamming and colleagues point out that the real potential of logging technologies is tapping into episodic memory, because "*a memory prosthesis should provide: information in a form which allows people to retrieve past episodes; information about many different episodes and the temporal relationships among them*". The memory prosthesis project also emphasized blurring the distinction between searching and browsing, which is common to Activity Trails as well. People tend to get lost or lose incentive when faced with large quantities of data, but an episodic view of a large data set should allow people to "*switch freely between constrained pattern-matching and free-range browsing*". The combination of these two characteristics is part of how this dissertation work defines episodic view of personal information.

The MyLifeBits project was one of the first ones in this area to continue the pioneering vision set by the Memex: we want to record everything about our lives and be able to easily access it when we need it. The MyLifeBits system consisted of a large database containing images, documents, phone conversation, emails, and anything else that could be digitized and stored, and that represents a person's life. In this sense Activity Trails shares the same vision and concern about the ever growing pervasiveness of digital media in our lives; but Activity Trails' emphasis is on *episodic* access to our digital lives, something that has not been addressed in many other systems so far and that reflects an important aspect of how we use the past in our everyday lives. The MyLifeBits system for instance, let users create custom tags that can be assigned to multiple items in very flexible ways. User tags have since become prevalent in how we organize, share and access our information, specially with the advent of new sharing services like Delicious and Digg, but also because they have become an integral part of searching and bookmarking. But tags do not represent an episodic view of our data, rather they represent an improvement over a rigid schema, allowing arbitrary labels and cross-membership of categories, but still only supporting semantic categories. An episodic view, in contrast, should present a history of episodes portraying a view of our interaction over time with any given number of items.

The MyLifeBits system also introduced a navigation technique called "pivoting", originating from database technology but later adopted by other HCI systems [17]. Pivoting allows partial faceted search within a database by progressively narrowing the result space based on user input and providing partial views of the data. This approach tries to blur the distinction between searching and browsing, which is closer to the episodic view that Activity Trails is trying to achieve as well. In other words, searching in episodic memory is partly a reconstruction process, where the recollected product comes about through a combination of what we know about ourselves and our past and what we actually remember directly or by association.

There has been little empirical evaluation of any of these systems however. One exception is recent work by Sellen and colleagues [68]. Using a personal recording device called SenseCam, they were able to measure recall performance using personal life-log images. In their study, images were recorded either at random intervals or manually initiated by the participants. They tested for the recall of discrete items over a period raging from three days to four months. Overall their results indicate that personal images do enhance memory performance, but that memory is more stable over time when recalling items by *knowing* about them rather than *remembering*. That is, when participants were asked if they "knew" a given event had occurred as opposed to actually "remembered" it happened, they seem to be able to recall more robustly over longer periods of time. Sellen et al. conjectured that this effect was due to a combination of inference, semantic information, and episodic memory, and they suggest that it constitutes an interesting advantage of using life-log images for designing support systems.

More recently, Sellen and colleagues [44] have began to tease out the impact the different life-logging technologies have on memory. Using a combination of SenseCam and GPS technology they conducted a user study where participants recorded their daily activities for a period of two consecutive weeks. Participants were interviewed five weeks later about their recollection of the events that took place during the recording phase. Using a combination of images, maps and verbal cues, Sellen and her group measured the quantity and quality of the items recalled as compared to “organic memory”. Importantly, they discriminated in their data between items remembered by true recollection (i.e., re-experiencing the event) and those inferred (i.e., recalled by reconstruction, based on what participants knew or guessed about their own activities) from the cues available. Their study suggests that different cues have different effects on the memory processes involved in recollecting the past. Images on the one hand seem to trigger direct recollection of memories, by promoting re-experiencing of events through episodic memory, whereas more abstract data such as physical location tend to promote inference, where people recall events through an active process of reconstructing the past from poorer perceptual cues but richer autobiographical knowledge.

One final area of research that is related to my work is that of user modeling based on activity recordings. In particular Horvitz and colleagues conducted a series of diary studies with the goal of building computational models of user behavior that can support work resumption and memory [22][42][28][39][21]. They showed that mining people’s calendars for salient personal events, or tracking changes in patterns of windowing usage are valid signals for detecting salient moments in activity. These pattern-based cues can be used when retrieving personal data, or can be used to detect moments where activity is disrupted. This shares common ground with the approach I am taking in my research (see Summarization on Chapter 4). In particular, Horvitz and Iqbal [42] found that the rate of certain operations increases significantly right before a person switches to attend to an interruption triggered by an email or instant message alert. So for instance, they found out that people tend to speed up their rate of typing or issuing menu commands in preparation to attend to an alert, possibly because “*users typically prefer to complete conceptual and/or motor subtasks before switching*”. I believe this type of windowing patterns are rich sources of signals for parsing activities on the desktop.

3 Field Work

3.1 Field Site

The field site for this study is an Immigration Law firm located in the San Francisco Bay Area. The firm is medium sized and its members have worked together as a team for almost a decade, starting as employees of a large immigration law firm before leaving to work at the current office. Shaped by their history of working together they have well established work practices for assisting clients with immigration issues. The division of labor at this office involves two main echelons: paralegals and attorneys. The role of attorneys is to address legal questions from the paralegals and clients, and to keep up with legal research, the industry, and government legislation. Paralegals' main job is to compile cases for filing. Compiling cases means gathering all the necessary documentation from foreign nationals in order to fill out the required forms for each visa category. It also includes drafting petition letters and following up on filing deadlines, which entail sending notifications to clients. In addition, paralegals are responsible for entering all this information into an online database.

In an Immigration Law firm billing is done on a filing basis. Unlike other types of law firms that have a need to keep track of billable hours, immigration billing is done on a per visa or work permit category basis. Every paralegal keeps a set of case logs with the latest status for each client and the corresponding follow ups. The emphasis of their work is on "timelining" rather than "time-tracking", that is, keeping a representation for each client case that allows them to track progress and have an understanding of to-do tasks. The nature of billing and the type of law shapes the work practice here: a common pattern emerges where everyone usually works on multiple cases simultaneously, since there is no need to keep track of billing hours and interaction with clients is done asynchronously through email; the only exception is when a case is being compiled prior to a filing deadline or when a new case is opened and substantial data entry is needed.

While the job responsibilities of paralegals and attorneys differ, everyone uses the same set of applications. This is a common feature of most mid-size to large corporate settings, but the collaborative nature of this workplace is benefited by the use of common tools sharing a common information server. The tools used by both paralegals and lawyers are: Microsoft Outlook as the email client, Microsoft Explorer for Internet access (including an online client database) and Yahoo Messenger (IM) for inter-office communication. Paralegals are expected to be able to take on any case in the office if needed. For this reason all communication with clients, documentation and forms for a case are entered in a common online database. All attorneys and paralegals share case logs and attorneys in addition have access to everyone else's email folders. IM is used as an informal means for sharing case knowledge. Workers use IM for communication rather than walking or interrupting someone else regarding a case. IM has become such an integral part of the work flow at this office that it is usual for workers to have IM windows open at all times.

There are two main roles for the use of IM. The first one is for paralegals to be able to quickly query colleagues regarding a case, and to get legal responses from the attorneys. The second one is for attorneys to be able to check with paralegals regarding their interactions with cases, since they are not directly involved on a daily basis like paralegals. It is very hard for paralegals who already have a heavy load of cases to annotate *all* the details in the database. Therefore a significant portion of a case's history is in fact *tacit-knowledge* available only through inter-office informal IM communication.

3.2 Data Collection Methodology

Researchers from many disciplines are taking advantage of increasingly inexpensive digital video and storage facilities to assemble extensive data collections of human activity captured in real-world settings. Hollan and Hutchins [37] have argued that the ability to record and share such data has created a critical moment in the practice and scope of behavioral research. The use of video recording for the study of workplace activity is increasingly central to advancing HCI research [66]. Lately, workplace ethnography has incorporated the use of computer screen recording [75]. Methodologically, this work falls somewhere in between elicitation and diary studies [16], and it stands in direct contrast to shadowing methods.

In the shadowing method [54], researchers follow participants throughout the day taking notes and embedding themselves in the workplace. Though the shadowing method provides the advantage of real-time recording and experiencing of the data, it has a high impact on the observer effect, that is, on influencing or distracting the worker from his or her normal behavior. Elicitation and diary studies by contrast try to minimize the intrusive factor [22][16] but the data obtained contains possible "noise" due to informants trying to explain their behavior in retrospect.

My approach shares a similar goal with diary studies in minimizing the observer effect, with the difference that I recorded behavioral data in real-time. I installed the Activity Trails Logger in the computers at the office and let the recording occur remotely, without having to be present. After a week of continuous logging, I collected the data and used it to manually build summary video clips. With these summaries at hand, I returned to the office a week later to interview my participants. There were a total of 6 participants, two attorneys and four paralegals, and the interviews consisted of one hour sessions where participants sat at their own desks, watched recordings of the past and narrated to me what they saw. This configuration is shown in Figure 3.1. Figure 3.2 shows what participants saw on the screen. The thumbnails on the top of Figure 3.2 correspond to separate three-minute episodes, and the bottom section is the activity playback. The complete six interviews were recorded on video and transcribed for analysis.

Figure 3.1: Interview setup. Participants sat at their workstations watching video clips of their own computers in the past, while narrating what they saw.

The scenarios in this thesis contain excerpts from these interviews, as well as descriptions of the activity on the screen as obtained from the original recorded data (e.g., pre-interview). In addition to free narrative, I also questioned participants in order to keep them engaged or whenever they disclosed potentially interesting details about their work. In general, there was no fixed script for these interviews. This recording/interviewing cycle was repeated three times over the course of one year. For each interview cycle, the time elapsed between recording and interview dates was extended, from one week to two, to one month. The idea was to first learn the details about the work, and then learn about the effects on memory of watching images further back in time. I also spent a considerable amount of time doing my work at the office, interacting with paralegals and attorneys, getting to know their work by direct participant observation and informal conversations.

My approach also shares similarities with the Experience-Sampling method [10][5] (a kind of elicitation protocol) where users are prompted by a computing device to log their experiences throughout the day. Activity Trails automatically

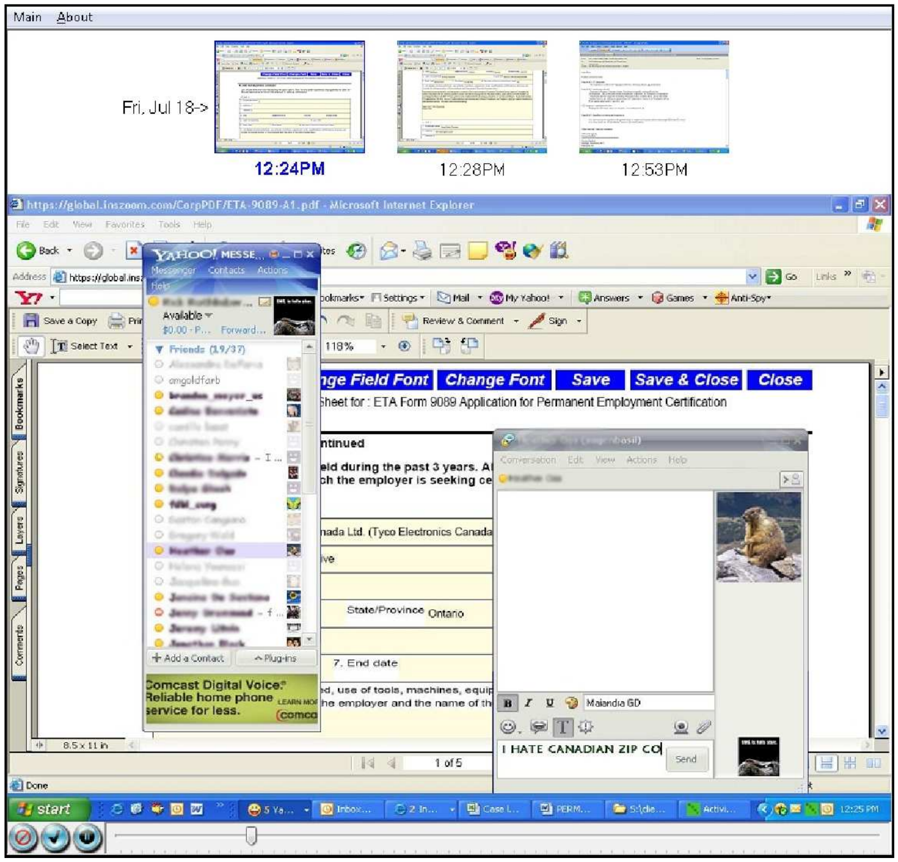

Figure 3.2: Activity summary used during interviews. Top thumbnails are individual clickable episodes. The bottom area is the playback showing a database window in the background and instant messaging in the foreground.

prompts workers to log their activity, in addition to continuously recording their work. The goal was to obtain references in the data to specific work moments that could be used later as anchor points for the interviews. The method removes the subjective component of data selection present in the shadowing technique, and in addition provides a richer and more complete record than the diary method. One of the drawbacks of screen recording is the potentially serious intrusiveness and privacy concerns. What I show through my work is that these potential difficulties are offset for the researcher by the insights gained from the richness of the resulting data.

3.3 Field Data

3.3.1 Timeline Representation of Context

As mentioned in the introduction to this chapter, paralegals keep case logs as a means to get a quick overview on the latest status for cases. This is a way for them to understand what needs to be done next, pending issues and follow ups with clients. But in order to actually address any client's action item, whether through email communication, phone calls or drafting letters, paralegals need to first put the case into context. The context for a case is the history of the cases' development, in terms of the documentation collected, forms and letters drafted, filling deadlines met and any past issues from interacting with the client. Context therefore is a chronological sequence of events, and what paralegals call "timelining". The following excerpt from a paralegal, illustrates this idea

> *"...sometimes you want to timeline with them versus just telling them so that they know that you've been timely so there's the email (pause) meaning that if we tell him that we got the signed form and that we filed the same day kind of just to let him know that it was all timely done versus say that we just filed it (pause) also for him to know that we didn't sit on it you know that we got the signed form today and that we filed today..."*

In order to build this timeline, a paralegal has to assemble the pieces of interaction and documentation history from several different tools. The database contains all dated documentation and forms, but it does not present this information in a manner that facilitates construction of a timeline. The following excerpt illustrates the use of the case log

> *"...that there is my case log but a lot of times we use it for the spreadsheet for the client or like if we get something that is a spreadsheet like an lcs spreadsheet they are all done that way, yeah i mean the (database name) is kind of too big (interruption) so this case log pretty much parses it down i personally have it at my end but is helpful cuz every so often...*
>
> *...generally is more or less to keep track of ok i gotta sign a case today so i keep it there and every week or once ever two weeks i update out forms for signature hmmm case filed today and then i move it to a file case thing and then every two to three weeks i meet with M (attorney's name) and we go through and say ok this case is filed and this has a (inaudible) not*

> *everything makes it on there (pause) it feels nice when i have it because then it's more organized but i have all that stuff in my head...*"

Each piece of information in the database is displayed through a different page so that workers have to switch back and forth in order to get a complete view. Paralegals also maintain an extensive set of folders in their Outlook's inbox, organized by company name. They search and browse this repository of archived emails in order to get the history of communication and interactions with the given client. They open several email threads to read and get a "refreshed" look at a case before they reply to or call a client. Figure 3.3 shows the construction of this timeline representation. The actual timeline at the bottom of the figure is an abstraction of the mental representation that paralegals put together from the three different tools used (shown in A, B and C). The purpose of this mental representation is to represent the *process* and not just the outcome of the interaction history with a client (the outcome can be directly obtained from the case log or a database note).

3.3.2 How Workers Build Context

Answering a client email or drafting a follow-up is a sample of a common work instance at this law office. It is how both attorneys and paralegals spend most of their time day to day. A given paralegal often has a long list of emails waiting in his or hers inbox at the beginning of a day, spanning back several days, as shown in Figure 3.3. Often an email window remains open in the background for an extended period of time, while the worker compiles and reviews all the necessary information about the case in order to draft a reply. This is what we call "building context" for a case, and as described in the previous section consists of putting together a mental timeline representation of the chronology of events for a case. Since all the information required to build this timeline is distributed across a variety of digital tools, and as shared knowledge among the workers, this process can take a while. Consider the following excerpt from an interview describing such an instance:

> "*...this is a client writing with a question and I am responding to the client but I think what am I doing let's see I probably don't know all the*

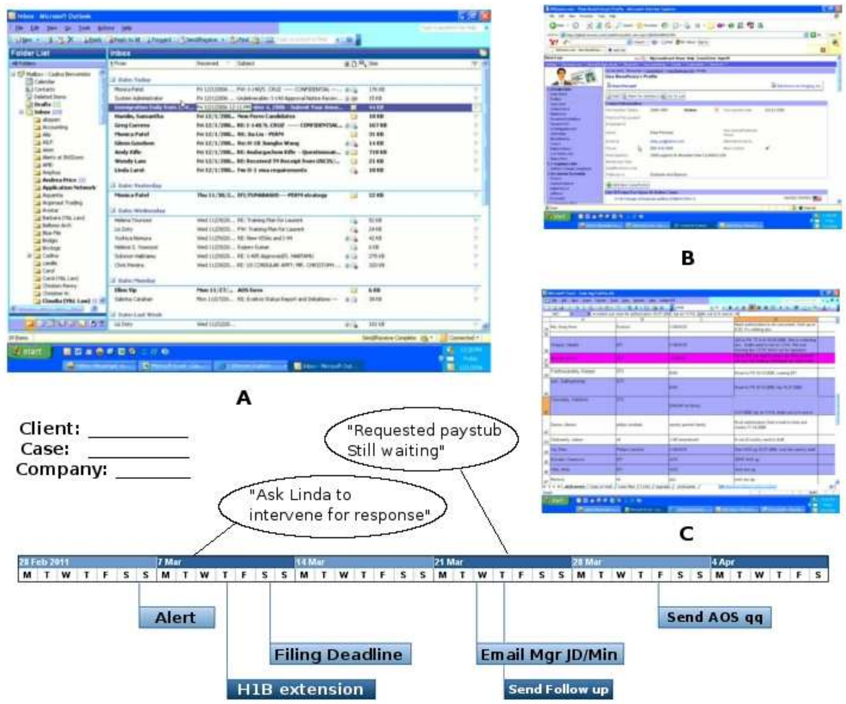

Figure 3.3: Timeline representation of a case history, assembled from several different applications and documents: A. Outlook's inbox, spanning several days, as well as a long set of folders, storing previous correspondences filed by company name. B. Web-based immigration database, containing filing documentation, deadlines and notes on conversations with clients. C. Case log spreadsheet, containing the latest status on the most recent cases.

> *information usually before I have to respond to a client I have to contact via IM another legal assistant which is what I seem to be doing*
>
> *. . . yes so I can't respond to their email and in the meantime I looked at a different email relating to our client possibly being acquired and I go back to the (client's name) email (short pause) the client email"*

The excerpt above comes from an interview with an attorney. In the recordings (not shown) she has a full-screen size window containing a reply to an email. There is no activity on that email but another smaller window (containing an IM) sits on top. She is typing a question to her paralegal there. This kind of self-initiated interruption is very common in my data. Sometimes attorneys have multiple IM windows open simultaneously. Because she does not get a response from the paralegal right away, she continues with something else by opening another email to read about a different client. At some point the screen shows her Outlook inbox, where a preview of the email she is reading is visible just below the list of incoming emails. The previously pending email is still open in the background on a different window. After a few minutes, she finally gets a reply from her paralegal, so she switches back to the pending reply window to start typing a response. Because this is an attorney, she finds it easier to just ask a paralegal for an update on a case before replying, using instant messages. She also could have gone into the database and browsed the entries for this client, or look at the case log for the paralegal working on this case (these logs are shared for this very reason). In this particular case though, the use of IM makes the process of context building a distributed social process. This complex social process is poorly supported through conventional tools. The Activity Trails prototype presented in the second part of this dissertation partially captures this social process, so it can be used as support. The final sections will describe a future implementation of the prototype where activity information is shared through a cloud process, making it a more suitable solution for this type of support.

As technology develops and new tools are introduced into the workplace, we see new work patterns emerge. People co-opt their environment and their tools in ways often not foreseen by their designers. For instance, the pervasiveness of email and the increasing sophistication of email clients like Microsoft's Outlook, have turned emails into important information repositories and tools. For the workers at this law firm, entire histories of interaction with clients are stored into carefully organized email folders. The following excerpt illustrates this

> *"... YES very often very often through the ... hmmm... through the (short pause) or through the folders we search a lot we search a lot (pause) I'll search through the folders and like I'll start to looking through eh the subheadings. Once in a while I search by ... if it's not too general I'll do the actual search . like a search through like please find eh (short pause) and sometimes i do subject . like I'll search by subject like so like that one I put in XXXX (client's name) cuz that's the client"*

Paralegals try to improve the process of context building by cutting and pasting the content of emails into the database. They also try to manually enter notes on the particular "issues" of each case. Workers try their best, but it takes considerable effort to be consistent and so not all nuances about cases can be directly recorded. Databases are not usually designed to directly support viewing histories, whereas email threads inherently contain that visibility. As described in the excerpt above, workers search through their email folders, but my data also shows that very often they also *browse* though these folders to be able to "view" multiple emails or conversation threads. Paralegals and attorneys therefore are constantly juggling back and forth between their email application and their database in order to completely build the context needed to address a case. This takes considerable effort since the information is *visually fragmented* across several windows and interfaces. But even though the database alone falls short of being a good repository for histories of interactions, it seems to be the first place workers go to when building case context

> *"...so we keep it in XXX (database name) so I cut and pasted to make that it's in XXX (pause) it's pretty much open all day ah by accident sometimes i click out of it (pause) there's a milestone emails and it keeps a history there it doesn't really come through outlook so (short pause) it's easier you end up cutting and pasting into XXX anyways and then all the milestones are there so when you click..."*

The tools available to the paralegals and attorneys at this office do not directly support the process of building context. The database and email application on the one hand, do not facilitate viewing the chronology of interaction in the history of a case. The fragmentation of a case's background information across multiple tools on the other hand, prevents access to the perceptual cues that can help trigger episodic memory for recollection.

3.3.3 Context Reinstatement

As I discussed earlier, my data shows that the dialectic between technology and work practice produces new work patterns that span across the use of multiple tools. This limits the ability of workers to use any single tool to effectively re-establish the context of past activity. We saw how paralegals have to juggle between their emails and the database in order to bring the history of a case into perspective. Because of this increased cognitive load, workers often resort to asking their colleagues for a refresher on a case. My data also shows a very interesting effect. Because of conducting my interviews using visual summaries from desktop recordings, I found many instances where a phenomenon called "context reinstatement" occurs. Context reinstatement is the restoration of the mental context present during the original episode, experienced as if "being there" by participants. This phenomenon is common part of our everyday lives, where we often experience vivid and sudden recollections triggered by images, our surrounding or smells. It is an important part of the episodic memory system[18].

In the physical world, the manner by which we organize and recall many items in our lives relies on natural cues, such as time, physical proximity, perceptual and mental associations. We sample the world continuously through perception, and leave physical traces behind through our actions. These traces acts as perceptual memory triggers that help us travel the path into and out from memory [35]. Episodic memory is a cognitive process that helps us recall specific instances of the past and the context in which they were experienced, and to preserve the temporal coherency in how we recall our personal lives [77]. In the digital realm however, these vivid recollection experiences are limited. This is mostly due to the fragmentation that occurs with all the different tools we use to get our work done, but also because computers only record the consequences of our actions, not their history. The visual summaries of desktop activity I employed, contain a temporal continuity that spans across application boundaries, and therefore contain the perceptual cues necessary to trigger vivid recollections of the past through episodic memory.

The following interview excerpts illustrate instances where context reinstatement occurs. I am using two signals to identify these: one are references to the physical context *beyond* what is visible on the screen; the other one are shift to the present grammatical tense, signaling a "being there" moment. In the case of descriptions containing details outside of the screen, these are usually triggered by an image actually *on the screen* (e.g., the name of a foreign national), or by a sudden stop in the flow of images in the summary (i.e., activity freezes). The following excerpt is an example of this type of indicator. It belongs to a paralegal. After quietly viewing the images on the screen for a few seconds, we see her pointing at the screen at the same place where she was previously typing an email

> *"...see I sent those questionnaires right before then (pointing to the screen) ...so looks like I got status of a case (long silence) this looks like I am looking for something I am looking for a memo or something that's why I am going through all these emails I think I was looking for this I140 thing yeah so all these emails here all this searching going all the way back (scrolling back in time on the screen) it's me searching (silence) so this guy was out of time (pointing to the screen) so then we found out that his EAD was approved so I was looking for this memo ..."*

By pointing to the screen, she is indicating that the names and words in the email trigger her initial recollection about the case. After a few more seconds of viewing the images, she scrolls back in time, and then forward again, as the details of this episode seem to come together in her mind. It seems as if the scrolling action is what gives her the temporal cues necessary to start building the context of the episode. She explains that she was searching through her emails for a memo regarding a client. As she explains this, more and more details seem to come about the case (i.e., that is was about an I140 form). Finally, she recalls details about the foreign national in question, including the background to his case that lead to the very moment portrayed on the screen. She then explains why she was searching (i.e., that his EAD was approved). I consider this an instance of context reinstatement because the images on the screen seem to first tap into her episodic memory recall, and then slowly bring back to the present the *complete* context of the episode, including details well beyond the contents of the screen. This context construction process is an integral part of the episodic memory system as described in Section 2.3.

par

The second excerpt belongs to the same paralegal. In this case, she is looking at a screen with no activity (paused suddenly after a few seconds of preview). The sudden pause seems to trigger something in her memory

> *"...I think I stopped in between here to ask XXX (paralegal's name) a question because of this conservative date (pause) yeah yeah I think I was working on four or five emails as they pop in (gesturing an incoming wave with her right hand) (long pause) I remember walking in there to ask her about this conservative date because it's shorter than (inaudible)..."*

Since there is no activity on the screen as she narrates, we have to assume that it was the break in the activity (the pause) *pause* that triggered the start of her recollection. She says she actually "remembers" walking into the attorney's office at that point. I consider this an instance of context reinstatement because she recalls the *physical* aspects the episode. This physical context is completely beyond what was captured on the desktop recordings. Interestingly, her context reinstatement is triggered not by any image on the screen, but by the temporal cues available in the series of images in the summary. In other words, it seems that the triggering cue is the *break* in the temporal flow. She seems to recall the entire physical and mental context of this event as if she was re-experiencing it.

The second type of signal that I am using to identify context reinstatement is a change in the tense of the narrative. Whenever tense shifts, it signals a shift in *cognitive perspective*. In language, tense is one of the main grammatical systems responsible for portraying the quality of mental events through discourse [31][71]. Discourse has a canonical structure where the order of events is portrayed through consistent use of tense, thus creating coherency in the narrative. When a shift in tense occurs, it signals a change in deictic center, for instance from narrative time to speech time (i.e. now). In our data, participants shift the canonical narrative tense to express what they remember as if *being there*, in effect bringing their past into the present through use of context-rich episodic memory. The following interview excerpt corresponds to another paralegal, and shows this effect. We use bold and capital letters to show emphasis in the speech

> *"...sometimes I have to figure where i left off, and I am like should I do something new? but then should I wait for her and then you kinda go back (pause)* **NOW I AM** *emailing on so i got a fax i emailed XXX (attorney's name) or IM'ed XXX about the fax and* **now i am** *emailing them saying (pause) call to discuss the fax"*

This paralegal is describing in general terms how she manages her work, not referring to anything in particular on the screen. She describes her strategies for resuming work after being interrupted while waiting for a reply. All of a sudden she sees new images on screen that somehow remind her of receiving a physical fax. At this point she switches to present tense in her narrative as she raises the volume of her voice to exclaim that "NOW I AM" emailing a client. At this point, the sequence of events in the episode start to unravel. She then recalls the details of what lead to and follow the email she was drafting, including the physical components of it (the fax). Again, I consider this an instance of context reinstatement because the participant recalled items outside of the information visible on the screen, and more importantly, seemed to have re-experienced the original episode by expressing a vivid recollection of the events.

The richness of the details obtained from these context reinstatement moments lead me to design the Activity Trails prototype. Initially, these desktop recording images were intended as an enhanced research methodology, but I soon discovered that they also facilitated occurrences of this phenomenon. This largely inspired the design and implementation of the work presented in this thesis. The tool I developed (described in the next two chapters) is intended to be a way to further explore the nature of context reinstatement on computers, as well as its applicability for everyday use. I see the use of tools like these as a novel approach to design and support how we view and manage our personal information in digital devices, one that is closer to how we *remember* our lives.

4 Prototype Implementation

The overall architecture of Activity Trails has the following design goals:

- Separate recording portion from graphical user interface for cross-platform portability.
- Maintain low memory and CPU cycle overhead to ensure that research can be conducted on the lowest common denominator equipment and with minimal impact to normal machine operation.
- Generate “raw” data in order to have flexibility for research.
- Prevent loss of data and ensure safe recovery from unexpected system breakdowns.
- Implement program to run as background process so that users perceive it as a “widget”, with no window frame in its graphical user interface.

Activity Trails was designed to run as two separate processes, interacting in a master-slave fashion, where the front end process (Graphical User Interface) is responsible for launching and monitor the back-end recorder of activity (Logger). The Activity Logger is a background process, with no user interface, that records all user activity in a continuous stream. It utilizes the built-in callback mechanism of the Windows operating system as well as multi-threaded worker processes to ensure minimal disruption to performance. The front end interface program was build using the QT[1] cross-platform development environment. It features a set of frameless windows that float on top of the computer desktop giving the sense to

[1] `http://qt.nokia.com/products`

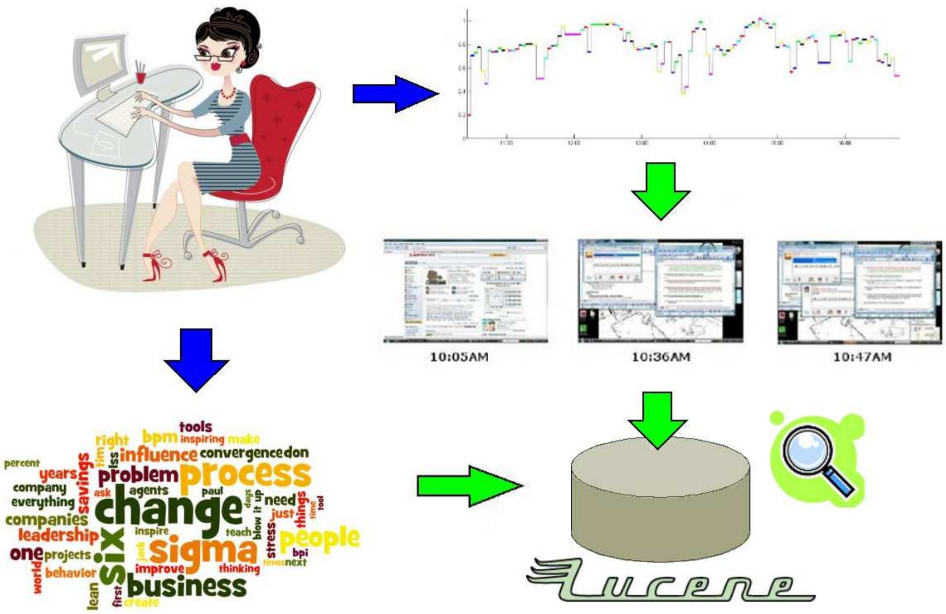

Figure 4.1: Logger for Activity Trails. Two streams of data are recorded: mouse and window activity data in the form of time series and word content from keyboard and applications. The time series data is processed to produce visual summaries of the most interesting moments of activity. The word data is indexed by *time* to enable search results representing *instances* in time where search terms occur.

the user of a widget that lives outside of any application's window frame. This interface was also designed to respond to direct keyboard input as a linguistic command line [61].This provides immediate response to the user, and allows for a windowless program, akin to Google Desktop or Spotlight.

4.1 Activity Logger

The logging and recording portion of Activity Trails was written to run on the Microsoft Windows 32-bit platform. It has been tested on Windows XP, Vista and Windows 7. It was developed using a combination of C\C++ languages using both the Win32 API and the Microsoft Foundation Classes library. Two additional libraries were used to improve performance: Microsoft DirectX10 and

The Independent JPEG Group image library. The basic design goal behind the development of Activity Trails was to produce a research tool that would allow recording remotely and continuously on low-end desktop computers (typically seen in offices), without impacting the performance of running applications as well as minimizing visibility. Activity Trails therefore was required to have a small memory footprint, perform data capture as close as possible to the hardware device level, and run on the desktop as a background process so that users would not notice it. There were several commercial options to capture the computer screen, such as Camtasia and Morae. These tools record video which has a higher overhead and is more difficult to parse, but they also produce output in proprietary Adobe Flash format, which is also more difficult for analysis. My decision therefore was to write a small and fast tool that could capture data with a low footprint and in a format that was easily readable, whether it be a database or an external analysis program, like Matlab or R. The running version of Activity Trails, has a memory footprint averaging 15MB and produces flat XML files for easy access to data. Appendix A shows the overall architecture of the logger.

Activity Trails takes a complete sample of the entire desktop at regular intervals. It does so asynchronously through the built-in callback mechanisms of the operating system (thus reducing performance impact). A "frame" of the desktop (i.e., a complete snapshot at a given frequency) consists of:

- Mouse coordinates
- Keyboard buffer
- Window geometry (position and size)
- Window state (active focus, minimized, etc)
- Window title
- Window application
- Window Content (including tabs and panes)
- Bitmap of the desktop

The output produced are flat XML files and JPEG images. XML is a flexible and portable format for the data. The structure of the XML data are long series of *frame* entities, each one corresponding to a timestamped snapshot of the desktop. The bitmap files for each desktop capture are named by their frame time. For instance, a frame capturing the desktop on March 18^{th} of 2009 at 6:48:24pm, would correspond to an image named `AT_screenshot_03-18-09_18-48-24.jpg`. The corresponding frame entity in the XML file would be as follows:

```
<frame date="03-18-09" time="18:48:24">
...
</frame>
```

Figure 4.2: XML Frame Output

Each frame XML entity contains a set of children with all information about the window and input device configuration at a given time. So for instance, if on March 18^{th} at 6:48:24pm the desktop contained two opened windows, one being a browser for reading Yahoo Mail, the other one being a text document called "workCompleted.tex", the XML output would look as follows:

```
<frame date="03-18-09" time="18:48:24">
<desktop numWins="2">
<window active="yes" exefile="iexplore.exe" module="IEFrame" port="80" rect="-8,1408,-8,1028"
server="us.mc431.mail.yahoo.com" title="(32 unread) Yahoo! Mail - Windows Internet Explorer" type="http"
url="/mc/welcome?.gx=0&amp;amp;amp;.rand=1odr727jhs0st"><![CDATA[Yahoo! Mail Welcome, Gaston!  You have 32 unread
messages: Inbox (1), Cogs102a (1), COGS102B (6), CogSciUCSD (1), context-rich-research (2), people (2), photos (1), receipts (9),
Scuba (6), user_accounts (2), YLaw (1)  Top Stories World NKorea threatens to attack US, SKorean warships (AP)  AP - North Korea
threatened military action Wednesday against U.S. and South Korean warships plying the waters near the Koreas' disputed maritime
border ...]]></window>
<window exefile="gvim.exe" minimized="yes" module="Vim" rect="-32000,-31840,-32000,-31975" title="workCompleted.tex +
(C:...edu_proposal) - GVIM"></window>
</frame>
```

Figure 4.3: XML Window Output

In the XML snippet above we see that the browser window (IExplorer) is visible (attribute `active="yes"`), but the document window ("workCompleted.tex") is minimized (attribute `minimized="yes"` and window rectangle has all negative values). We can also see that the content of the Yahoo Mail window is included in the data (everything inside the `CDATA` element). There is also information about the window positions and geometry, titles and URLs. Similarly, information about the mouse and keyboard is structured using XML entities. In the listing below we can see that the mouse was at location `x="306" y="144"`, in a 1024 by 768 pixel display (`<out right="1024" bottom="768"`). We can also see that the user had typed "*did you go to your meeting*". The keyboard buffer contains all typed characters since the last frame, as seen in the `keyBuffer` node below

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!-- Output file for Activity Trails Program v0.5.5 -->
<!-- Start: Thursday, March 18, 2009. 14:57:04 Pacific Daylight Time -->
<out right="1024" bottom="768" keyrate="4" framerate="4">
...
<frame date="03-18-09" time="18:48:24">
<mouse x="306" y="144"></mouse>
<keyBuffer><![CDATA[did you go to your meeting]]></keyBuffer>
<desktop numWins="3">
<window ...
```

Figure 4.4: XML Input Devices Output

XML output file size is limited to 1MB, so the program produces multiple XML files which ensures fail-safe recovery in case of computer malfunction. Screen capture images are compressed using the JPEG codec averaging 150KB for a 1024 by 768 32-bit screen resolution. The program also does cleanup of these external files, to keep file storage size low, as described in the next section.

4.1.1 Data Processing

The data produced by the Activity Trails logger is "raw" data. This was a design decision made because of the research nature of the tool. By raw I mean data representing direct measurements of the recordings, such as window geometries and mouse coordinates, instead of aggregations. Other loggers, such as Microsoft's Vibe Logger [40] aggregate data into events such as windows closing, opening or resizing. This raw data gives me the flexibility as a researcher to aggregate events in multiple ways during off-line post-processing in order to better explore the data space. The raw data recorded is treated as a statistical time series, since the data is sampled at regular intervals. All observational variables are discrete values: mouse coordinates, keyboard count, window geometry and overlap.

Finally, Activity Trails performs an analysis of the recorded data processed in the form of a time series, in order to identify recurring patterns. These recurrent patterns represent *structural saliency* in the data that we can use to parse it for summarization purposes. I consider transition points around recurring patterns as "breakpoints" in the activity (with the assumption that a recurring pattern will be something familiar to the user). Summarization of activity in this case means to separate continuous activity into discrete segments or "episodes". The program uses these segments to create summary files, which are an XML specification for sequences of images. The GUI portion of Activity Trails (see Section 4.2) interprets this XML specifications as visual summaries playbacks. Each summary is composed of a series of episodes, each episode is a sequence of static images of the desktop (see Section 4.1.1). Because Activity Trails records static images every `N` seconds, the playback mechanism inherently produces an additional form of summarization. Images are played back at 1 frame per second (fps), producing therefore a time compression of $\frac{1}{N}$. For instance, a 3 minute summary episode (playback time) represents 15 minutes of actual recorded activity. The rest of the images are deleted from disk to preserve space (otherwise the program can store up to 1GB of data daily). This assumes that our parsing algorithm would have selected the best images (the most interesting moments) from the data, something that will eventually approach validity once a more advanced algorithm is in place (see Chapter 6).

Third-party libraries:

- **GSL** (GNU Statistical Library), used for processing time series [2].
- **CLucene**, a C language implementation of the Lucene Search Engine by the Apache Foundation[3].
- **TinyXML**, a parser by Lee Thomason[4].
- **SAX** (Symbolic Aggregate Approximation) by E. Keogh et al[5].

[2] `http://www.gnu.org/sofware/gsl/`
[3] `http://clucene.sourceforge.net`
[4] `http://www.sourceforge.net/projects/tinyxml`
[5] `http://www.cs.ucr.edu/~eamonn/SAX.htm`

- **WordSpy** (A Steganalysis-based Approach for Genome-wide Motif Finding), by G. Wang and W. Zhang[6].

- **GNU Aspell** (A Spelling Suggestion Library[7].

Time Series Smoothing

In order to process the raw data output from the logger as a time series, meaningful variables need be computed and smoothed over a sliding window. Smoothing is the process of calculating the running average over the time series. A running average for a time series t is computed as follows

$$s_i = \frac{1}{n} \sum_{j=i}^{i+n-1} t_j \tag{4.1}$$

where t_j is the time series sample at time j, s_i is the running average at sample i, and n is the length of the sliding window. I chose $n = 90$ empirically to be a suitable value from my data. Given that the logger captures frames every 3 seconds, 90 sample points correspond to a 180 second window.
The variables computed for the time series are as follows:

- Mouse Displacement (amount of mouse change between frames)

- Total Window Count (all windows opened, visible and not visible)

- Visible Window Overlap (a percentage of all visible windows, explained below)

- Window Switching Count (how many times the active window changes, this variable is NOT smoothed, it is the total count within one sliding window)

Mouse displacement is computed as the Pythagorean distance traveled by the mouse from frame to frame (i.e., capture points), normalized by the size of the screen's diagonal

[6] `http://cic.cs.wustl.edu/wordspy/`
[7] `http://www.aspell.net`

$$\frac{z_n - z_{n-1}}{z_{max}} \tag{4.2}$$

where $z = \sqrt[2]{x^2 + y^2}$, $z_{max} = \sqrt[2]{screenWidth^2 + screenHeight^2}$, and x, y are the horizontal and vertical mouse coordinates respectively.

The window overlap variable represents the percentage of all visible windows that are occluded. It is computed such that a maximum value is obtained when one maximized window covers the entire screen and the minimum value when all windows are minimized

```
FOR each visible window
IF current window rect > previous window rect then
SELECT largest window
COMPUTE union with previous window rect
END FOR
DIVIDE largest window area by union area
```

Figure 4.5: Pseudo-code for Computing Window Overlap

The time series is normalized to have a mean of zero and standard deviation of one

$$\frac{s_i - \bar{s}}{\sigma} \tag{4.3}$$

where $\bar{s}$ is the sample mean and σ is the standard deviation. Figure 4.6 shows a sample of the smoothed time series. This sample shows data for an entire day of recording, roughly from 9am to 5pm, as displayed in the x-axis. The y axis is the normalized values. We can readily see, for instance, that this user has several overlapping widows on her desktop most of the time (third row from the top), except for transitional moments where she might move things around on the screen.

Summarization

My summarization approach is based on the *semantics* of the data, in contrast to other approaches based on image statistics such as video key frame extraction and summarization[23]. The underlying principle is that people have different *styles* of working with tools, and that these styles reflect *syntactic* patterns in terms

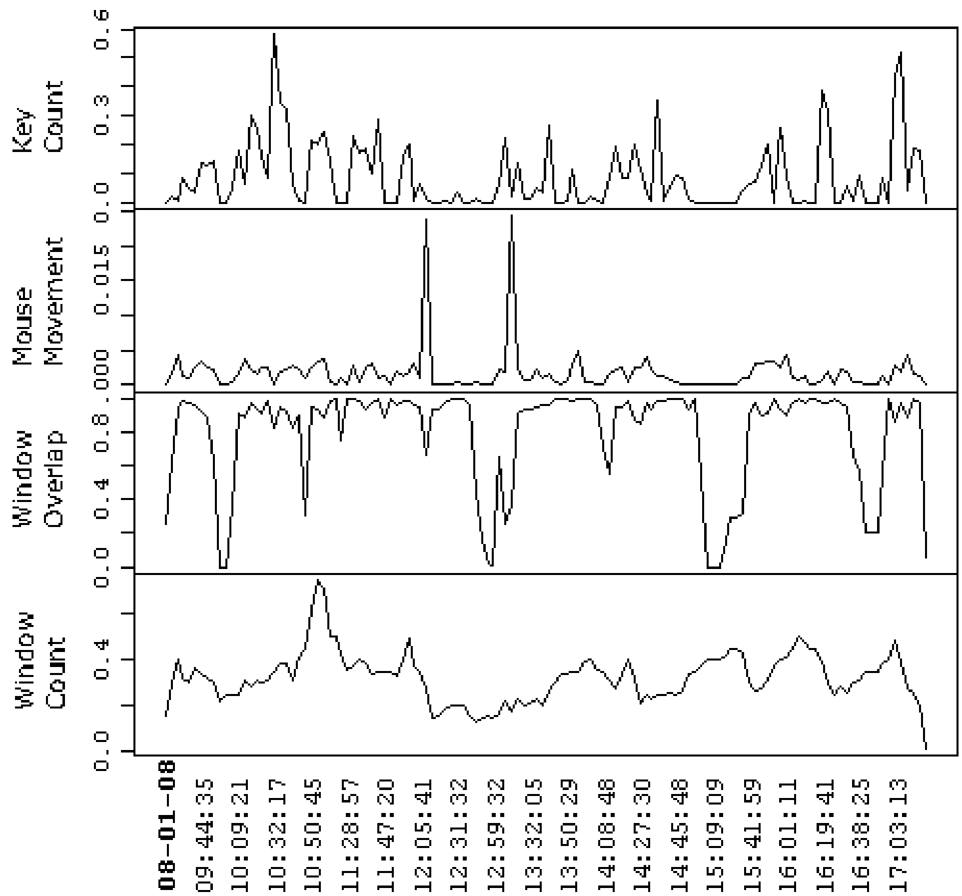

Figure 4.6: Running averages, each point representing 3 minutes of activity. Plot shows an entire day of recording.

of the visual arrangement of windows on the desktop, the type of applications they prefer to use and how the use them in tandem. These preferences and the patterns of "windowing" styles that emerge are stable for a given person in a given computing configuration. They originate from attentional and perceptual constrains based partially on the limitations of the screen real estate but also on the different performance profiles of users. This notion is the basis for my algorithm. Take for instance a person who might prefer to maximize his windows to cover the whole screen for visibility and then toggle the other windows in the background using the task manager (Alt-Tab on Windows) or clicking the icons on the taskbar. Another person in contrast might like to have multiple windows visible at all times, and position them next to each other to improve visibility or use multiple tabs or panes within a browser or application. Yet another person could employ both of these styles of windowing, but use them exclusively for different types of work (e.g., checking email and Facebook versus programming). These windowing patterns are usually stable for the same person on the same computer performing the same type of activity. They are contingent on the type of attentional focus preferred by the individual or demanded by the task at hand (i.e., single vs. multiple attentional focus or multitasking)[13].

There are also patterns which are common to all users and all configurations, such as patterns of input behavior. If a person switches between windows of the same type on a regular basis and with a regular frequency, it indicates a relation in activity between those windows (a similar notion was studied in [59]). Similarly, when a person changes the rate of typing on the keyboard, or switches from mostly moving the mouse to mostly keyboard typing, it might indicate the beginning or resumption of some particular activity (a similar finding was obtained in [42]). My approach aims to detect patterns such as these, but not to use them for identification purposes, rather as salient points from which to select images for summarization. The question remaining is how exactly does the algorithm identify these points along the time series.

In a current NSF proposal [37], Hollan and colleagues propose to explore the application of several computer vision and data mining methods for this same purpose. In particular I am interested in exploring the use of the Symbolic Aggregate Approximation method (SAX)[43]. SAX is a novel and elegant approach which produces a symbolic representation of a time-series, reducing its dimensionality and numerosity and making analysis more noise-tolerant. Figure 4.7 shows a visual representation of my data through SAX. My algorithm employs these symbolic approximation as a preliminary step in detecting salient points in the activity.

The principle behind the SAX algorithm is to first reduce the dimensionality of the time series using the Piecewise Aggregate Approximation method (PAA) by diving the time series into `n` equal slices, and approximate the data in each

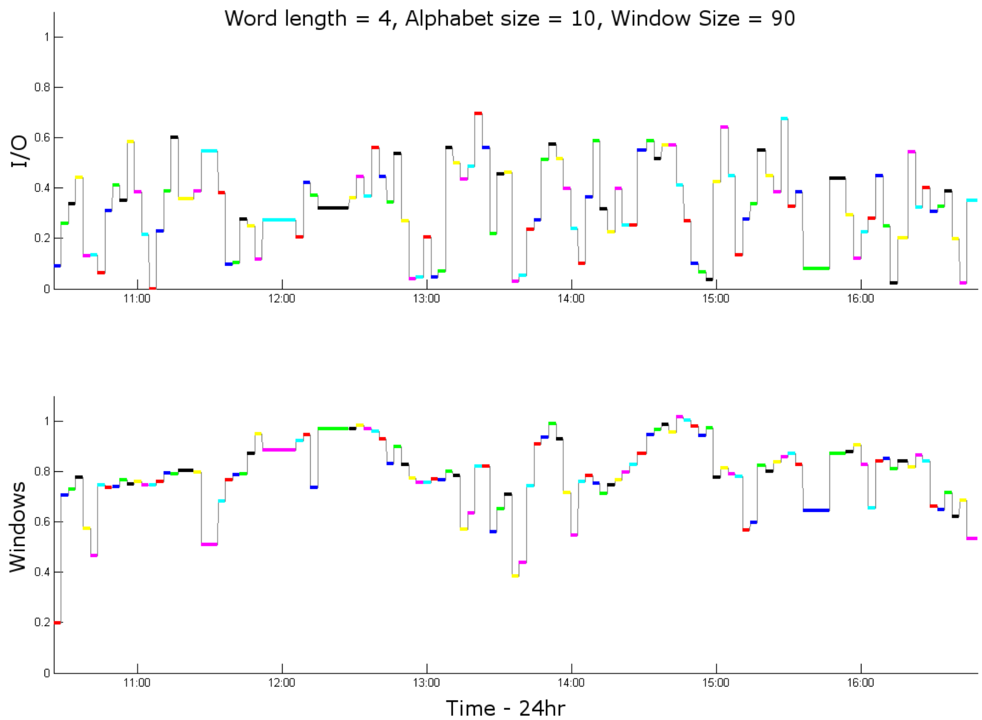

Figure 4.7: SAX representation of my data. A time series is converted into a string of symbols *bdfjdgibbcejigfa*, for an alphabet size of 10 and a word size of 4. The upper plot represents an aggregate of keyboard and mouse data, and the lower plot is window variables (i.e., overlap, visibility and switching. Each flat segment in the plots represents a four-letter string or symbol, in the form *bdfj*

frame by a linear function (i.e., the small flat lines in Figure 4.7). The second step in SAX is to discretize the PAA reduction into symbols. Using the notion that any normalized time series has a Gaussian distribution, the algorithm obtains equiprobable points in a Gaussian and uses them as thresholds to determine the assignment of symbols. Each symbols is assigned to an alphabet letter, and then groups of symbols constitute words of a given length. For instance figure 4.7 shows data from one of my participants, for an entire day of recording. SAX was run on this data using a sliding window of size 90 (i.e., the "slice" size) with an alphabet size of 10 letters and a word size of 4. Since my data was sampled at an average of 3 seconds apart, the 90 sample window corresponds to 180 seconds of activity (3 minutes). The alphabet size is the maximum allowed by the algorithm and the word and window lengths were determined empirically. In other words, 90 points in my data were "compressed" into 1 symbol using 4 letters (or 4 linear approximations). The resulting SAX representation consist of a long series of letters in the form `bdfjdgibbcejigfa` where the first four letters `bdfj` constitute a symbol representing the data from `10:25:13` to `10:28:17` on July 30^{th} of 2008.

The main advantage of a using a symbolic representation like SAX is that the resulting lower dimensional data space can be used as an approximation to greatly simplify the computations involved. In addition, and as discussed by the authors[43], symbolic sequence matching techniques can be used to identify *motifs* efficiently in large data sets. In the field of genetics, a sequence motif is a nucleotide or amino-acid sequence pattern that is widespread and has, or is conjectured to have, a biological significance. For my purposes a motif is simply a recurring pattern that is statistically significant against the background time series. The onset of a motif represents therefore a salient point in the data. My approach is to consider those salient points as *breakpoints* to be used for activity summarization (i.e., points at which the program samples images for display).

Once the logging data has been converted into symbols, the second step of the algorithm employs a tool called WordSpy to identify recurring motifs. WordSpy is based on Steganography, which is a technique for concealing information by embedding messages in background covert text[79]. I chose this tool because it can be used with a any dictionary of symbol letters, instead of strictly aminoacid sequences. Dictionary means that the program builds grammars using Hidden Markov Models for both the background words and the motifs, and then progressively learn a series of models that are most likely to have generated the sequence of motifs. Within the realm of Bioinformatic tools that can extract structural information from sequences, WordSpy falls in the category known as probabilistic optimization[25]. The main advantages of WordSpy over other tools in the same category is that it is less likely to be trapped in local minima. This is because WordSpy's approach is a combination of word counting and statistical modeling approaches, so it is less likely to detect spurious motifs than the pure word frequency approaches. Overall, WordSpy is a fast procedure for discovery and identification of the *complete* set of significant motifs present in the data.

The output from WordSpy contains dictionaries for all word lengths up the requested word size, and for each length the set of significant motifs with their frequencies:

```
-------------------------- Significant Motifs ----------------------
Motifs Zscore NZscore Gscore Freq(%) Occur# Seq# Wordlist
--------------------------------------------------------------------
[ motifs of length 2 ]
[ motifs of length 3 ]
IGD 6.4 null null 0.89635 5 1 igd
DGJ 4.4 null null 0.91614 3 1 dgj
ADI 4.3 null null 0.91295 3 1 adi
[ motifs of length 4 ]
AEGI 25.2 null null 1.25601 5 1 aegi
BCFJ 19.8 null null 0.94441 4 1 bcfj
DBFJ 16.2 null null 0.63013 3 1 dbfj
IHCB 10.0 null null 0.93822 3 1 ihcb
[ motifs of length 5 ]
HHGAE 29.9 null null 0.63340 2 1 hhgae
IJFCB 29.0 null null 0.63384 2 1 ijfcb
BDGII 21.8 null null 0.63376 2 1 bdgii
[ motifs of length 7 ]
FJJGDAA 489.8 null null 0.63450 2 1 fjjgdaa
HHGAJFB 274.3 null null 0.62570 2 1 hhgajfb
[ motifs of length 8 ]
IGDABCGJ 838.4 null null 0.63457 2 1 igdabcgj
```

Figure 4.8: Sample Output from WordSpy

The motifs (of length equal to the word length selected for the SAX conversion) are then located in the time series and treated as breakpoints. Each breakpoint is the start of an episode in a summary. For instance, if motif `aegi` occurs as the third word in the SAX sequence, it's onset is located 6 minutes from the start of the time series (since each word corresponds to 90 samples, which are equal to 180 seconds or equal to 3 minutes). If the time series begins at `10:25:13` then there will be an episode starting at `10:31:13` in the form of `<episode date="07-30-08" time="10:31:13" duration="180"/>` in some summary XML file for playback in Activity Trails. Figure 4.9 illustrates this.

Textual Content and Indexing

In addition to the time series data described in the previous sections, the logger also records textual information such as the words being typed on the keyboard and the title and content of visible applications and documents. This information

```xml
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!-- Summary specification file for ActivityTrails v0.5.5 -->
<summary transition="3">
    <episode date="08-01-08" time="10:05:32" duration="300"/>
    <episode date="08-01-08" time="10:36:35" duration="300"/>
    <episode date="08-01-08" time="10:47:55" duration="300"/>
</summary>
```

Figure 4.9: Visual summary data representation. Each image corresponds to a thumbnail of an episode, which can be played back at full screen. The XML is the file specification for the summary as produced by Activity Trails.

can generate a model of commonly used words which can be used to find interesting *moments* in the histories of activity. By interesting we mean moments where what was being read or typed is relevant to the user based on their own criteria. Since people are very accustomed nowadays to searching the desktop or the web, I included a search engine in my design. I am employing a C-language implementation of the popular Lucene search engine. I chose this engine because it is very fast and simple to deploy as a static library. The program treats the textual context in each captured "frame" as a separate *document* in the search index. An indexable frame therefore (equivalent to an indexable document in search engine vocabulary) constitutes a collection of terms (words) obtained from an XML frame in the output data files. When searching on this index, the user will in effect be searching through *histories* of past activity, as opposed to collections of static documents (e.g., emails, word documents or websites). This is a subtle yet crucial distinction with conventional desktop search. It will become more evident as I discuss how people interact with the graphical user interface (Section 4.2 and Chapter 5). This is the basis for studying how to support *episodic* access to personal activity. In other words, when a user enters a series of keywords for search, the result yields not only documents, emails or web pages that contain those keywords, but also their

distribution through time. One can think of search "hits" as instances withing a chronology of events, where an event is the occurrence of keywords in documents. This interaction was designed to support the use of episodic memory by providing exposure to the temporal distribution of events.

As each XML frame with the recorded data is processed into a time series, the textual content is saved into plain text files for indexing. Each file contains three categories: window titles, visible window content and keyboard input. Each category is used as separate field for indexing (fields can be used as filters for searching). Below is a sample of one of these text files:

```
Keyboard: hi just got your  order i am out  of town right now  so the earliest i can  put the book in the mail is sat you

Titles: Gmail - Inbox (1) - gastoncangiano@gmail.com - Mozilla Firefox Gmail - Inbox (2) - gastoncangiano@gmail.com - Mozilla
 Firefox Gmail - Inbox - gastoncangiano@gmail.com - Mozilla Firefox Yahoo! - Windows Internet Explorer context.tex
 (C:) - GVIM context.tex + (C:) - GVIM

Content: Make Y! your home page Get Internet Explorer 8 Denied a Bank Account? Get Account Now Prepaid VISA. Yahoo! Sear
ch Web Images Video Local Shopping More Answers Directory Jobs News All Search Services Advertising Programs Disable Assi
st Search: No suggestions. Please begin typing search query. Search Assist: On | Off Popular Properties Yahoo! Home My Jun 1
, 2009 Page Options Color:Ocean Tangerine Violet Oyster Grass Pink Switch to narrow layout Want more ways customize pa
ge? Try ...
```

Figure 4.10: Sample Output Text File for Indexing

The text content from visible application windows exploits the Microsoft Active Accessibility API (http://msdn.microsoft.com/en-us/library/ms971310.aspx), which provides access for visually impaired users. One external text file is created per every frame that differs at least in **one character** from the previous one. This selection process produces redundancy, but it is necessary because duplicate frames represent distinct *instances* of an item across time. The search results are spread out during the process of creating visual summaries. For example, whenever instances are contiguous, only the first one is selected. Also, if we get three results within one hour of activity, since each visual episode represents fifteen minutes (compressed to three minutes of playback time), they would ideally need to be five minutes apart from each other.

The keyboard input stream is recorded as raw data, that is, every single key press is recorded. So for instance, if the user types the sentence "I'll call you back" and then hits the `ENTER` key, the output captured is `SHTi'll call you backRET`, where `SHT` and `RET` are the shift and enter key respectively. Similarly, if the user forgets to type the last letter "l" in the word "call" and then corrects it, the captured data could be `SHTi'll cal youBCKBCKBCKBCKl you backRET`, where `BCK` represents each press of the backspace key. This raw stream gets processed into readable words by replacing each meta-character (non-printable character) into its function (i.e., back spaces are processed by removing them and deleting the last character from the stream). Shift keys are only processed when they are used to print alternative characters, such as "@", or "}", otherwise they are ignored since all words are converted to lower case when indexed into the search engine. Carriage returns are processed as white space. The program later identifies word boundaries using white space, punctuation marks or symbols. Since the keyboard is a continuous stream, the program also timestamps characters. These timestamps are used to determine whether a word was cut out during the frame capture process or if the user slowed down or interrupted typing. Word boundaries are also determined through use of these timestamps.

Finally, I am using a very popular spelling library called GNU ASPELL. One of the advantages of this library is that in addition to common typos, it can detect "sounds like" errors, making it a superior spelling suggestion engine. Words smaller than three characters in length (e.g., articles, prepositions or key shortcuts) and numbers, are removed from the stream prior to spell checking. While spell checking any given word, the suggestion with the smallest number of differing characters is the one picked as a replacement. The program also keeps a dictionary of all the words that have been replaced together with their replacement in an external text file. If a word is found in this dictionary, it will not be replaced. It is more likely that a personal noun (like the name of a product) will appear repeatedly than a typo. Below is an example of this dictionary file, where we can see that a word like "augniscient" which is the name of a company will not be replaced more than once, and that a word like "sophisitcated" which is a typo does get replaced by the proper spelling. Other words such as "summarization" which is not in the dictionary but appears repeatedly in this user's vocabulary, is also only replaced once.

```
iinterprets interprets
treatmen treatment
summar summer
augniscient omniscient
geomtr geometry
ometry cometary
zahr zara
gastonhghh gaston
ithe the
nnnngg nonnah
screensize screens
logg log
hitch04 hitch
hutch04 hutch
fron freon
ymbolic symbolic
summarization symmetrization
zyzs zs
sophisitcated sophisticated
saliency salience
emph mph
ialgorithm algorithm
```

Figure 4.11: Sample Dictionary Text File

The final stage of data processing is entering the content into the search engine (i.e., indexing). Each document contains three fields: Titles, Keyword, and Content. Each string entered into these field is passed through a filter, called an "analyzer" in Lucene's terminology. We use the *standard analyzer*, which is a combination of a standard tokenizer, and standard lower case and stop word filter [8]. A *tokenizer* is a filter that splits strings at punctuation and white space characters, but recognizes email addresses and internet hostnames as one word. Filters remove words that are not useful for searching purposes, such as the stop words of the English language (e.g., "the", "a"). The following listing shows an example of the data parsed by this analyzer:

```
Pre-filtering: Gmail - Inbox (1) - gastoncangiano@gmail.com - Mozilla Firefox Gmail - Inbox (2) - gastoncangiano@gmail.com -
 Mozilla Firefox Gmail - Inbox - gastoncangiano@gmail.com - Mozilla Firefox Yahoo! - Windows Internet Explorer context.tex
 (C:) - GVIM context.tex + (C:) - GVIM

Post-filtering: gmail inbox 1 gastoncangiano@gmail.com mozilla firefox 2 yahoo windows internet explorer context.tex c cygwin
 home gaston edu docs context gvim
```

Figure 4.12: Sample of Filtering

Fields are indexed using the flag `TERMVECTOR_YES`, which specifies that the search

[8] `http://lucene.apache.org/java/2\_9\_2/api/all/org/apache/lucene/analysis/standard/StandardAnalyzer.html`

engine builds a set of term-frequency vectors. These vectors are used to obtain the set of terms (words) repeating with the highest frequency on a set of documents. Section 4.2 explains how these vectors are used to provide **suggestions** for searching. Searching for a single term on a week's worth of recordings data takes approximately 300 milliseconds. The result from this search query represents a series of timed frames, ranked by the engine's scoring algorithm, which it is based on document-term frequencies and term positioning within document (see http://lucene.apache.org/java/2_4_0/scoring.html).

4.2 Graphical User Interface

Based on the observations made from my initial field study at a law office, I designed and implemented a second generation Activity Trails prototype. The main objective was to allows us to study the use of an episodic interface for accessing everyday personal computer activity and contrast it with how people normally access their personal information. Following the guidelines about episodic interfaces presented earlier and the observations from our interviews, we implemented a design that emphasizes a visual representation of event chronologies that allows usesr to easily navigate by zooming in and out of episodes. As our interview data showed, users experienced the most vivid recollections of events while *navigating* visual previews of activity. That is, in addition to recalling events and information from the visible cues on the screen, users seemed to benefit from scrolling through images of their desktops and using thumbnails to jump forward and backward between previews. Participants consistently remembered more details as they took time to navigate back and forth through the images. Our design attempted to increase the ease of this interaction for users.

The other major design consideration was to provide a widget-like feel, moving away from the traditional framed application window. Haystack[45], PHLAT [20] and MyLifeBits [29], the works most similar to Activity Trails, all employ the traditional application window frame. One potential drawback of this is that users are led to view each as yet *another application* in addition to the many they

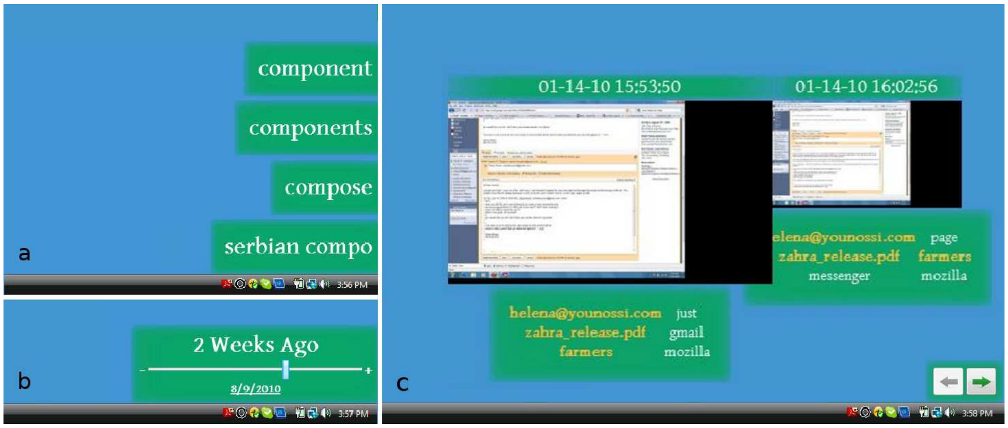

Figure 4.13: Graphical User Interface for Activity Trails. a) Direct keyboard input from user produces suggestions based on frequent terms. b) Date selector. c) Results are displayed as thumbnails floating on top of desktop representing a visual chronology of events. Users can scrub thumbnails to get a preview of the activity or double click them to see a full screen summary playback. Each thumbnail also displays the date and time of the event and a *keyword cloud*.

are already using to get their work done. This potentially increases task load by requiring them to add yet another tool to their repertoire. Our approach is to have the interface for Activity Trails be a widget that floats on top of the desktop as other tools such as Google Desktop, Spotlight for the MacOS, and Enso for Windows[9] do and with interaction directly via the keyboard input. Figure **??** shows this interaction.

The program normally runs invisibly as a background process, available via an icon on the taskbar. Once the program is activated and the user starts typing, a list of suggestions is available from the search engine (Figure **??**a). Once a set of query terms is entered, the user has the option to select a time range using a slider or a calendar (Figure **??**b). The slider provides a logarithmic time scale, so that users can easily scroll back from days to weeks to months from the present time. Once all information is entered, the program performs a search through an episodic index that results in a visual chronology of events (video clip thumbnails) representing every instance of the occurrence of the search terms (Figure **??**c).

[9]`http://www.humanized.com/enso/`

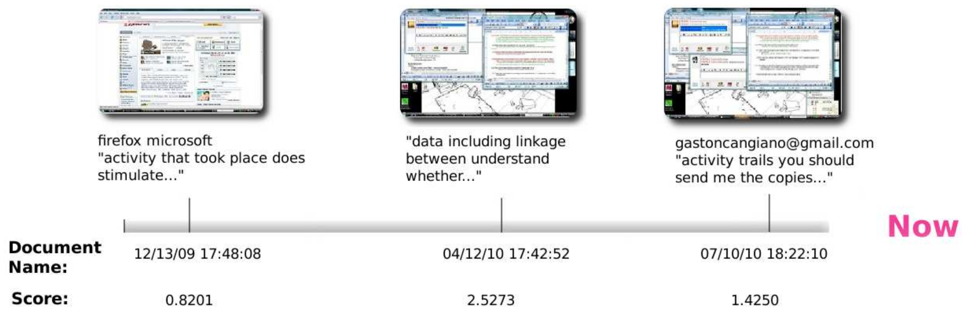

Figure 4.14: Search Results for query: +Titles: gastoncangiano@gmail.com +Keyboard:activity. Hits represent an ordered timeline, where each document is a collection of text content and a corresponding visual summary.

One critical feature of Activity Trails is that when users perform searches they are in fact searching through *instances of time* where the search terms occur. Consequently search engine *hits* are equivalent to a chronologically ordered list of matching instances of events in the activity stream. This yields series of visual summary clips (as shown in Figure 1.1) that represent a visual chronologies of events and allows easy navigation and browsing. Each thumbnail can expand to full screen size producing a“rewinding” effect on the desktop for users to re-experience the associated activity episode.

We define *episodic* access in the following terms:

- Visual representation of chronologies of events
- Ability to navigate sequence of episodes in order to stimulate the reconstruction of events
- Multi-scale playback of events in order to provide perceptual cues for episodic memory

The thesis validates its claim with the design of a prototype fulfilling the requirements for episodic access. Activity Trails is evaluated with real-life users through case studies in order to understand more about the nature of the research problem at hand and the extent to which the software is successful in providing support.

5 Case Studies

I conducted a usability study with the extended version of the Activity Trails prototype. The objective of the study was to get a broad sample of users, to collect detailed data on usage comparing access via episodic images with the tools people normally use to get their work done. We installed the Activity Trails Logger on the machines of participants and asked them to record their activity for a week. Participants recorded activity at their own location, usually their offices or homes, using a laptop computer. We conducted the post-recording interviews on-location whenever possible. During our interviews, the protocol consisted of first asking participants to go back to specific work instances using the tools they normally use, including paper notes or any other documents. We then asked them to use our prototype to retrieve other instances of activity so that we could draw a comparison in terms of the *type* of information they recalled. The program has an automatic prompt that asks users on a regular basis to enter typed notes regarding the activity they are performing. We used those log entries as reference for our questions during the interviews. The sessions lasted approximately one hour and were transcribed.

Our study had a total of ten participants. Half of these were researchers in Cognitive Science at the University of California in San Diego. Three were postdoctoral researchers, the rest graduate student researchers. The other half of the participants were undergraduate students at the same university, from a set of diverse departments. Two of the participants were female, and ages ranged from 20 to 40 years.

The results from this study break down into two categories: those who benefited from viewing images of their activity by recalling important details they

```
> Thu Aug 5 12:56:11 2010
I amrunning SOBI on subjects 34-38 to compare to ICA with Christina tomorrow
*********************************************************
> Thu Aug 5 14:08:09 2010
I am sending emails to the summer research undergrads i'm working with telling them what to
do for next monday's meeting
*********************************************************
> Thu Aug 5 15:08:39 2010
I am moving eeglab files to a flash drive and thinking about how to prepare for meeting with
Christina tomorrow
*********************************************************
> Fri Aug 6 15:11:16 2010
just concluded 2 hour meeting w Christina (comparing filitered and unfilitered data, ica and
sobi decompositions)
*********************************************************
```

Figure 5.1: User log showing dated entries based on periodic prompts by the Activity Trails program.

had forgotten and those who did not get any useful additional information from what they had already recalled using their own tools and notes. These two groups were independent of the type of work being performed, so there were researchers who did benefit from viewing images as well as non-researchers. This result means that the advantages of episodic interfaces could be independent of the *kind* of activity, but directly related to the value attributed to restoring previously gained knowledge and the characteristics of the tools employed. More specifically, our results indicate that for those activities that have a repetitive nature, such as data analysis or programming, images are helpful only marginally as a means to reflect on the overall work done and next steps. By contrast, those tasks that involve learning and acquiring new knowledge benefit greatly from viewing images and reinstating the context of past work in order to "refresh" memory. Our conjecture is that the utility of images stems from users' value attributed to retaining new knowledge gained in long-term memory. In addition, tasks that require or are usually performed with the use of highly structured time-based organization tools, such as versioning systems or journaling software, also do not benefit directly from the use of images. Our results therefore cut across both categories of participants. They give insight into the nature of those tasks that can benefit from the use of episodic images, regardless of the general type of activity involved. The following four sample cases illustrate these results by showing data from both types of activity (research and non-research), including one case where images triggered useful memory recall for each type.

Sample Cases on Research

Doctoral Researcher A.M. is a Ph.D. candidate student at the University of California. Her research uses electroencephalography (EEG) to investigate affective face processing with potential application to the study of deficits in empathy and social cognition in autistic populations. She is currently in the last phase of her dissertation research, analyzing data already collected using a variety of methods and comparisons. She organizes her work using a folders containing project files for Matlab, a statistical analysis tool, which she employs heavily for her data analysis. In addition, A.M. keeps a lab log book with her current progress and a list of to-do items. She also employs her email as a way to keep reminders about to-do items and pending issues, especially for meetings with her supervisors and colleagues. At the time of the interview A.M. was working on data analysis for a paper, but she was at a standpoint waiting for co-authors to come together to decide the typse of analysis and comparisons to perform. We interviewed her at the office, with her laptop and her notes.

In the first half of the interview, we asked A.M. to pretend that she had to continue working on her latest project. She proceeded to open up the corresponding Matlab files in the editor and consult her lab log book. She spread out her notes on the desk, which included printouts of EEG graphs and plots from her previous analysis. Once her environment was setup, she continue to check her email to find the last correspondence with her supervisor, who had suggested applying a new type of filter (FIR) to the data. It was still unclear to A.M. whether she should apply the filter before or after running independent component analysis (ICA and SOBI).

The second half of the interview consisted on having A.M. use the Activity Trails prototype to find more information about the project. After typing two keywords related to the project (SOBI and the name of her supervisor) three thumbnail previews appeared as the result. It wasn't immediately apparent to A.M. that she could scrub these preview thumbnails to view the activity, nor that

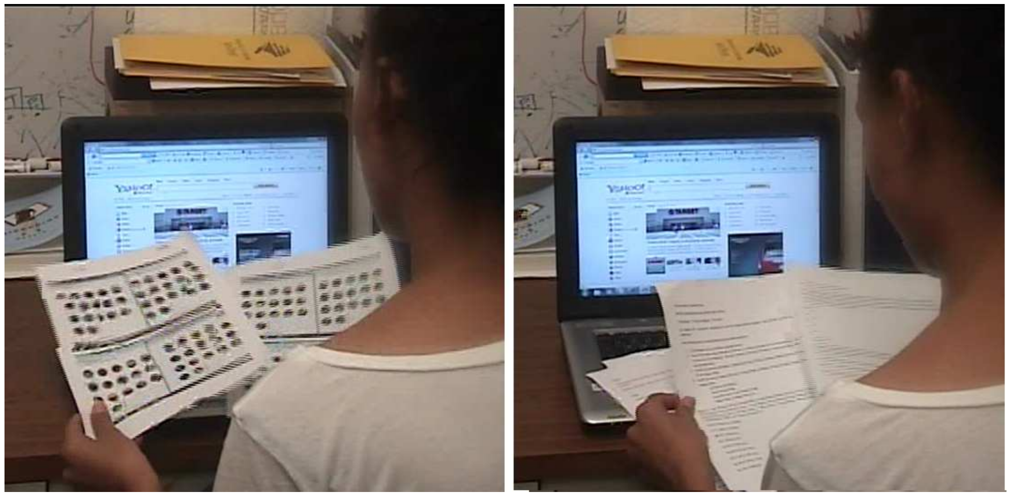

Figure 5.2: User's notes and printouts used during interview.

she could expand those into full screen view. Once instructed she took the time to view the summary. The first thing that came to her was a word document that she had put together for a meeting. In this document various EEG graphs are placed together for comparison purposes. She recalls the reasoning behind this document and the details of the meeting with her supervisor.

> *"...she recommended the FIR filter (pause) I asked her about her recommendation on filtering the data before or after EPIC'ing the data, and she didn't have a preference so I'll just continue doing it the way I've been doing it (long pause) but what I would like to do is find out what Word doc I am referring to here (long pause, while viewing images) ok, so that's what I did (pause) to compare the results from ICA and SOBI source separation ahmmm so that we can get a good visual display of all subjects (pause) the meeting was cancel so I had time to put this document together and bring it to the next meeting..."*

After a few moments of thinking, she recalls that during the meeting they had discussed doing a comparison with both types of analysis (ICA and SOBI) and that the application of the new FIR filter was still open to discussion, so that it would make sense for her to try applying it both before and after the analysis as a means of comparison. During the final section of the interview we asked A.M. about her general impression of the software. She replied

> *"...well I think that I am just at a stage where I have several things partly done and I have to decide what to focus next (pause) so it is useful*

> *for me to see all of my ongoing efforts ahmmm and that will help me decide exactly what to do next...*"

Overall, A.M. seems to have benefited the most from reflecting on her work, using a combination of images and the log. A.M. found the combination of the dated entries in the log and the images to give her a nice comprehensive picture of what she did, and where to go next.

Post-Doctoral Researcher D.G. is a post-doctoral researcher at the University of California. His research is on developing and evaluating new statistical analysis methodologies for electroencephalography (EEG) studies. More specifically, he is currently working on evaluating event-related potential (ERP) methodologies with the purpose of implementing a new toolbox to be made available to the research community. The recordings we used for the interview consisted of D.G. working on a journal paper where several different ERP methodologies were compared. The paper included simulation data, graphs and description of the coding done in MATLAB (a statistical and data analysis software).

D.G. journal paper is divided into multiple sections, each corresponding to a separate MS Word document. In addition to these documents, D.G. uses a journal software that has come to replace his lab notebook. This software allows him to create chronological entries that are digitally linked to actual documents on his hard drive or web pages. In this way, updates to any of these documents are automatically synchronized with this journal entries. In addition, a tagging mechanism allows him to easily search for images and text, and lets the program automatically categorize entries. This is very helpful because in addition to notes, he also cuts and pastes code from MATLAB into the journal entries. The only drawback to this software is that it does not support versioning of documents, since all reference to documents are links and not copies of older versions. He also maintains a lab wiki, with frequent entries regarding his algorithm development and techniques.

Once D.G. started previewing the images generated by Activity Trails, he explained how he starts the process of writing by first reading from his journal entries. He then proceeds to cut and paste the content from the journal entries into the corresponding Word document for the journal paper. He says that his starting point is usually the journal entries, and that he then expands the writing in Word but frequently consulting his simulation code and other resources such as the wiki. We see him on screen switching regularly between the journaler, Word and the MATLAB environment. When prompted during debriefing about the potential benefits of using Activity Trails, he said that he sees it as a good versioning revision system, since he does not usually keep older version of documents so it would be useful to be able to review previous edits of code or writing.

Sample Cases on On-line Search (Non-Scientific Research)

Undergraduate Student Performing On-line Learning J.A. is an undergraduate student at the University of California. He is an upper-division student in Cognitive Science and has recently finished a course on perception, which has given him the motivation to research the topic of cochlear implants. More specifically, he is interested in researching an idea to develop an implant based on a new approach using lasers. The interview covered one long recording session over the course of an evening, where J.A. spent several hours reading articles, blogs and wikies on the web. When prompted to summarize his session and resume where he left off, he proceeded to open a word document that he had put together the night of the recordings. This document was a detailed set of notes on the findings of his readings, plus annotations on to-do items, such as topics for further research. At the end of the document, he had clearly written a set of follow up items in order to know where to continue his research.

For the second part of the interview, we asked J.A. to view the recorded images from the Activity Trails prototype. After scrolling through these images, the first thing J.A. recalled was a wiki entry he made to correct someone else's answer to the topic of compound action potential (CAP). He had forgotten about this episode and thought that maybe he should go back and check on that entry again to see if it had been updated. Even though this was not central to his research, he was happy to remember that. The next thing he recalls is a related topic he researched on laser types based on frequencies. He had not written down any of the information he learned about these lasers so he was happy to see that he was able to recall some of the details of that search while viewing these images.

> *"...I remember researching dipoles but not understanding that very well (pause) ahmmm I still don't completely understand it but from what I mostly get is (pause) ahmmm oh I came across it because when you put ahmmm (pause)* ***OH NOW I REMEMBER*** *(chuckle) so I came across the word 'mid-infrared' as far as the frequency range they are using (pause) and because of this dipole moment principle you can put infrared light into a chemical and because of the movement of the particles..."*

J.A. commented during the debriefing portion of the interview that he thought the process of viewing these images was like doing "speed studying", referring to the ability to quickly review a previous study session. He was also pleased by the ability to remember things he had forgotten and that seem to be helpful for him to resume his research with a refreshed memory of what he did.

> *"...it's really cool to bring back those memories, like having a bowl of cereal that night (after a few minutes) it all of a sudden started jumping back to me it's nice (pause) it's like speed studying..."*

Undergraduate Student Performing a Job Search K.K. is an undergraduate student in mechanical engineering. He also works at a plumbing supplies manufacturing company where he models pipe fittings using computer-aided design software (CAD). His recordings consisted of a job hunting session, an activity he performs regularly. It consists of going through job listings, primarily in Craigslist, finding the most interesting positions and sending cover letters and resumes. This activity is usually performed all in one uninterrupted session. K.K. has been looking for a new job for a while, but he is still looking for the same type of position, except that he is interested in a company that offers a position and an environment for growth. He is specifically interested in a job that allows him to manage other people, do skill training and team collaboration.

The recordings during the interview show K.K. opening multiple tabs on a browser window. He has at least twenty tabs open at a time, one for each job posting worth reading. Once he finds an interesting listing, K.K. copies and pastes the title of the listing onto a Word document. He also includes the URL of the company if that is available. He relies on his memory to recall the details of the job description from the ads. The documents serves only the purpose of letting him know what applications he has sent out.

The first thing that K.K. notices while looking at the recorded images is that he had a large number of tabs open at a given moment and that it was extremely difficult to distinguish and make sense of them visually. He comments how this problem pushes him to quickly decide which tabs are really important and worth keeping open. He drags those tabs into separate windows to organize his search so that the important listing are easily identifiable. The images show K.K. switching back and forth between the browser windows, the Word document and GMail in order to cut and paste the job information and send out resumes and cover letters. Once this is done he closes the corresponding tab or window. He reflects on this process and realizes that he uses tabs and windows as a means of visually organizing his search so that he knows what is done and what still needs to be addressed. He then continues through the images recalling the one company he was most interesting in applying and that the position had been already filled. He reaches the end of the available images in the Activity Trails prototype.

6 Closing Discussion

Based on my initial field work at the law firm, I discovered a definite need for supporting the process of context building. This need arises from the fact that workers at this office have to employ a multiplicity of tools in order to put each case into context. The context for a case corresponds to the history of interaction and chronology of events associated with a particular foreign national. The background information for "refreshing" this timeline chronology of events is dispersed across several tool, such as a database, email client and spreadsheets. Using recorded images during my interviews, I saw how workers experienced vivid recollections of the past, and where able to restore the complete context of a case from the images. This data inspired the design of a second generation prototype of an episodic interface to past activity, which I called Activity Trails.

The case studies I presented in the last section were targeted to gain better understanding about the phenomenon of context reinstatement through the use of images and its application to everyday work through the use of an episodic computer interface. We saw on the one hand that for those tasks where participants had invested a considerable amount of time and effort in learning new concepts (such as the case for the undergraduate student learning about cochlear implants) gaining access to that information again using images was extremely useful. In particular, there was value attributed to being able to consolidate knowledge into long-term memory, as our participant explained with the expression "speed studying". Activity Trails supports the process of context reinstatement through a multidimensional interface that provides associative visual cues, exposure to the temporal sequence of events, and the ability to consciously reflect about past activity. This combination seems to be effective for restoring past context and helping users consolidate previous knowledge into the present. We believe Activity Trails should be applicable for many tasks sharing a similar need.

On the other hand, the nature of the tools involved in a task also had an impact on the benefits of images for memory. For the case of scientific researchers working on journal papers, I showed two cases that employed lab notebooks. One of them, participant A.M., used paper notes and printouts, and had to recur also to emails and browsing her computer environment in order to put herself "back" into the project mindset. With the use of Activity Trails images, A.M. was able to recall important details about the interaction with her boss and her research data and therefore figure out a course of action for her work. In contrast, the other participant (post-doctoral researcher D.G.) used a sophisticated journaler software to maintain chronological entries with live links to documents in his digital lab notebook. He did not recall anything particularly insightful that would help him resume his writing, although he was able to reflect on his methodology. Nonetheless, D.G. made a comment about Activity Trails being useful as a sort of versioning system allowing him to see previous unsaved versions of his documents. One could argue that A.M. could benefit from using software program like D.G. was using, but often for busy researchers investing that kind of time in organizing daily activity is just not their work habit. Activity Trails offers the ability to gain organizational knowledge in *retrospect* by reflecting on past activity, without the time and effort investment up front.

Based on the work presented here and the ongoing evaluation of the Activity Trails software prototype, I am very encouraged by the early signs of the potential of this new generation of episodic interfaces. The availability of large and inexpensive storage devices as well as increased processing power makes this a time of unique opportunity for researching life-logging technologies. I believe that the use of episodic access to personal information could be a valuable addition to a rich ecology of personal information management facilities.

A UML Diagrams for Activity Trails

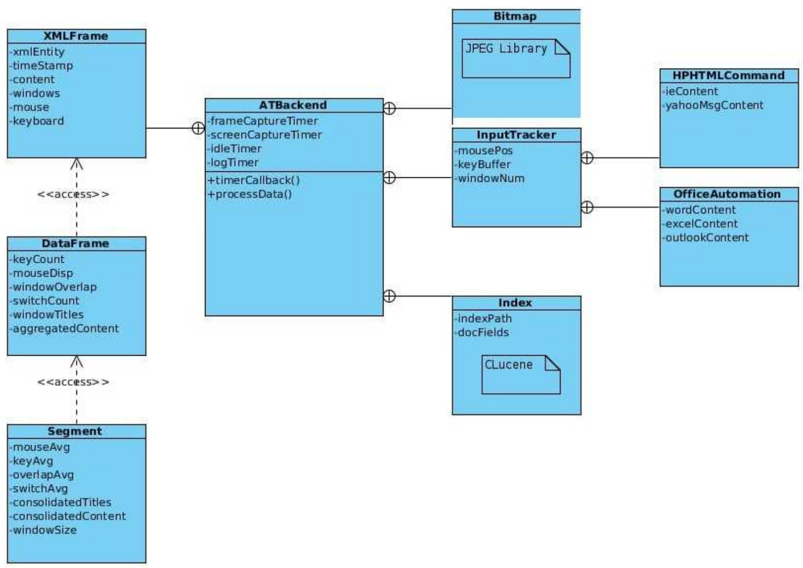

Figure A.1: Activity Logger Class Diagram.

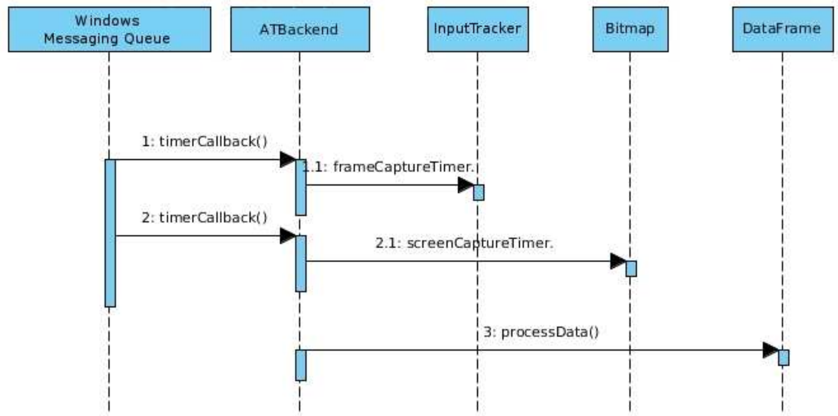

Figure A.2: Activity Logger Sequence Diagram.

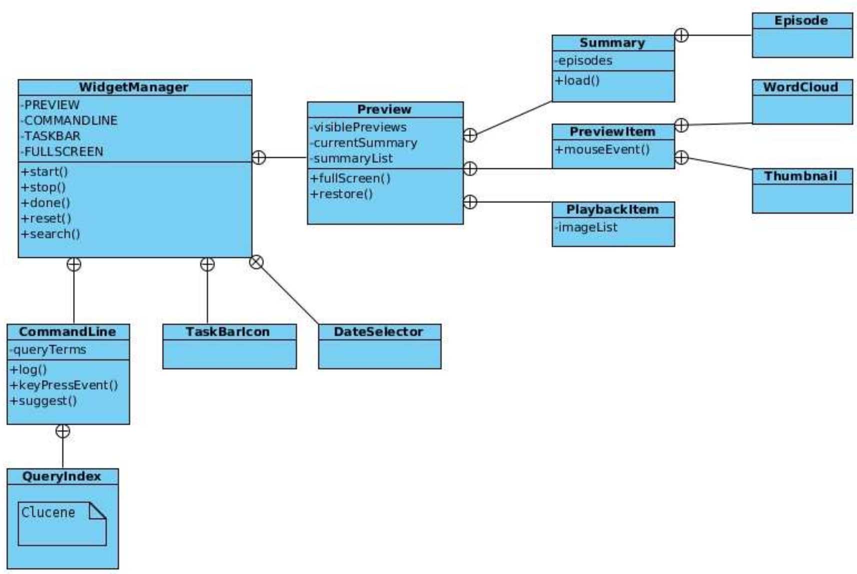

Figure A.3: Graphical User Interface Class Diagram.

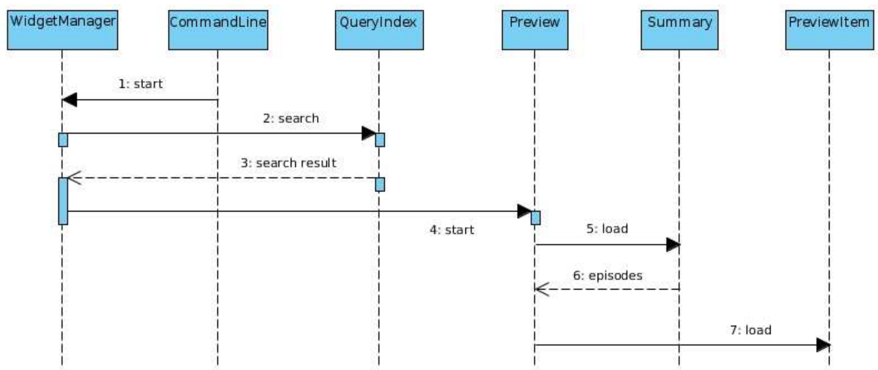

Figure A.4: Graphical User Interface Sequence Diagram.

Bibliography

[1] A. Baddeley. Working memory: looking back and looking forward. *Nature Reviews. Neuroscience*, 4(10):829, 2003.

[2] B. P. Bailey and S. T. Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Trans. Comput.-Hum. Interact.*, 14(4):1–28, 2008.

[3] J. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 211–220, New York, NY, USA, 2006. ACM.

[4] S. Barley. Technology as an occasion for structuring: Evidence from observations of CT scanners and the social order of radiology departments. *Administrative science quarterly*, 31(1):78–108, 1986.

[5] L. Barrett and D. Barrett. An introduction to computerized experience sampling in psychology. *Social Science Computer Review*, 19(2):175, 2001.

[6] L. Barsalou. *Remembering reconsidered: Ecological and traditional approaches to the study of memory*, chapter The content and organization of autobiographical memories, pages 193–243. Cambridge University Press, Cambridge, 1988.

[7] N. M. Bradburn. *The science of self-report: Implications for research and practice*, chapter Temporal representation and event dating, pages 49–61. Lawrence Erlbaum, 2000.

[8] W. F. Brewer. *Remembering. reconsidered: Ecological and traditional approaches to the study of memory*, chapter Memory for randomly sampled autobiographical events, pages 21–90. Cambridge University Press, 1988.

[9] D. Broadbent and D. Broadbent. *Perception and communication*. Pergamon Press London, 1958.

[10] P. Brown, K. Carter, M. Eldridge, M. Flynn, M. Lamming, G. Louie, G. Louie, P. Robinson, P. Robinson, and A. Sellen. The design of a human memory prosthesis. *The Computer Journal*, 37:153–163, 1993.

[11] A. B. Brush, B. R. Meyers, D. S. Tan, and M. Czerwinski. Understanding memory triggers for task tracking. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 947–950, New York, NY, USA, 2007. ACM.

[12] V. Bush. As we may think. *The Atlantic Monthly*, 1973.

[13] G. Cangiano and N. Tan. Investigating styles of window usage on desktop computers. Presentation at the Human Computer Interaction Laboratory, University of California at San Diego, May 2008.

[14] S. Card, T. Moran, and A. Newell. The model human processor- An engineering model of human performance. *Handbook of perception and human performance.*, 2:45–1, 1986.

[15] S. K. Card and A. Henderson, Jr. A multiple, virtual-workspace interface to support user task switching. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pages 53–59, New York, NY, USA, 1987. ACM.

[16] S. Carter and J. Mankoff. When participants do the capturing: the role of media in diary studies. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 899–908, New York, NY, USA, 2005. ACM.

[17] D. H. Chau, B. Myers, and A. Faulring. What to do when search fails: finding information by association. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 999–1008, New York, NY, USA, 2008. ACM.

[18] M. A. Conway. Episodic memories. *Neuropsychologia*, 47(11):2305–2313, 2009.

[19] M. A. Conway and D. C. Rubin. *Theories of memory*, chapter The structure of autobiographical memory, pages 103–137. Lawrence Erlbaum, 1993.

[20] E. Cutrell, D. Robbins, S. Dumais, and R. Sarin. Fast, flexible filtering with phlat. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 261–270, New York, NY, USA, 2006. ACM.

[21] M. Czerwinski and E. Horvitz. An investigation of memory for daily computing events. In *Proceedings of HCI 2002*, pages 230–245, 2002.

[22] M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 175–182, New York, NY, USA, 2004. ACM.

[23] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[24] S. Davies, S. Allen, J. Raphaelson, E. Meng, J. Engleman, R. King, and C. Lewis. Popcorn: the personal knowledge base. In *DIS '06: Proceedings of the 6th conference on Designing Interactive systems*, pages 150–159, New York, NY, USA, 2006. ACM.

[25] P. D'Haeseleer. How does dna sequence motif discovery work? *Nature Biotechnology*, 24(8):959–961.

[26] A. N. Dragunov, T. G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li, and J. L. Herlocker. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 75–82, New York, NY, USA, 2005. ACM.

[27] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i've seen: a system for personal information retrieval and re-use. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, New York, NY, USA, 2003. ACM.

[28] R. I. From, M. Ringel, E. Cutrell, S. Dumais, and E. Horvitz. Milestones in time: The value of landmarks in. In *In Proceedings of Interact 2003*, pages 184–191, 2003.

[29] J. Gemmell, R. Lueder, and G. Bell. The mylifebits lifetime store. In *ETP '03: Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*, pages 80–83, New York, NY, USA, 2003. ACM.

[30] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell. Passive capture and ensuing issues for a personal lifetime store. In *CARPE'04: Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences*, pages 48–55, New York, NY, USA, 2004. ACM.

[31] M. Gernsbacher and T. Givon. *Coherence in spontaneous text*. J. Benjamins, Philadelphia, 1995.

[32] G. P. Ginsburg and D. L. Smith. Exploration of the detectable structure of social episodes: the parsing of interaction specimens. *Ecological Psychology*, pages 195–233, 1993.

[33] V. González and G. Mark. Managing currents of work: Multi-tasking among multiple collaborations. In *ECSCW 2005*, pages 143–162. Springer, 2005.

[34] E. Hall. *The dance of life*. Doubleday New York, 1989.

[35] T. Helstrup, R. De Beni, C. Cornoldi, and A. Koriat. *Everyday Memory*, chapter Memory pathways: Involuntary and Voluntary Processes in Retrieving Personal Memories, pages 291–315. Psychology Press.

[36] J. D. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Human-Computer Interaction: Special Issue on Human-Computer Interaction in the New Millenium*, 7(2):174–196, 2000.

[37] J. D. Hollan and E. L. Hutchins. *Designing User Friendly Augmented Work Environments: From Meeting Rooms to Digital Collaborative Spaces*, chapter Opportunities and Challenges for Augmented Environments: A Distributed Cognition Perspective. Springer, 2009.

[38] J. Holquin. Generation M: Natural Multitaskers. Website, 2005. `http://www.cbsnews.com/stories/2005/05/10/eveningnews/main694344.shtml`.

[39] E. Horvitz. Learning predictive models of memory landmarks. 2004.

[40] D. R. Hutchings, G. Smith, B. Meyers, M. Czerwinski, and G. Robertson. Display space usage and window management operation comparisons between single monitor and multiple monitor users. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 32–39, New York, NY, USA, 2004. ACM.

[41] J. Huttenlocher and V. Prohaska. *Memory for Everyday and Emotional Events*, chapter Reconstructing the Times of Past Events. Lawrence Erlbaum, 1997.

[42] S. T. Iqbal and E. Horvitz. Disruption and recovery of computing tasks: field study, analysis, and directions. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 677–686, New York, NY, USA, 2007. ACM.

[43] L. J., K. E., W. L., and L. S. Experiencing sax: a novel symbolic representation of time series. *Data Mining Knowledge Discovery*, 15:107–144, 2007.

[44] V. Kalnikaite, A. Sellen, S. Whittaker, and D. Kirk. Now let me see where i was: understanding how lifelogs mediate memory. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 2045–2054, New York, NY, USA, 2010. ACM.

[45] D. R. Karger and D. Quan. Haystack: a user interface for creating, browsing, and organizing arbitrary semistructured information. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 777–778, New York, NY, USA, 2004. ACM.

[46] C. Kaufman-Scarborough and J. Lindquist. Time management and polychronicity. *Journal of Managerial Psychology*, 14(3/4):288–312, 1999.

[47] S. Knoll, A. Hoff, D. Fisher, S. Dumais, and E. Cutrell. Viewing personal data over time. In *CHI '09: Proceedings of the Workshop on Interacting with Temporal Data*, New York, NY, USA, 2009. ACM.

[48] R. Kraut, R. Fish, R. Root, and B. Chalfonte. Informal communication in organizations: Form, function, and technology. *Baecker (1993): Readings in Groupware and computer-supported Cooperative Work. Morgan Kaufman*, pages 145–199, 1990.

[49] H. Lee. Time and information technology: monochronicity, polychronicity and temporal symmetry. *European Journal of Information Systems*, 8(1):16–26, 1999.

[50] J. Liu, P. J. Hong, and E. R. Pedersen. Research trails: getting back where you left off. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1151–1152, New York, NY, USA, 2010. ACM.

[51] E. F. Loftus and W. Marburger. Since the eruption of mt. st. helens, has anyone beaten you up? improving the accuracy of retrospective reports with landmark events. *Memory and Cognition*, 11(2):114–120, 1983.

[52] L. Loukopoulos, K. Dismukes, and I. Barshi. *The multitasking myth: handling complexity in real-world operations*. Ashgate Pub Co, 2009.

[53] P. Luff, J. Hindmarsh, and C. Heath. *Workplace Studies*. Cambridge University Press, 2000.

[54] G. Mark, V. M. Gonzalez, and J. Harris. No task left behind?: examining the nature of fragmented work. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–330, New York, NY, USA, 2005. ACM.

[55] G. Miller, E. Galanter, and K. Pribram. *Plans and the structure of behavior*. Holt, Rinehart and Winston New York, 1960.

[56] M. J. Muller, W. Geyer, B. Brownholtz, E. Wilcox, and D. R. Millen. One-hundred days in an activity-centric collaboration environment based on shared objects. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 375–382, New York, NY, USA, 2004. ACM.

[57] L. Nadel, J. D. Payne, and W. J. Jacobs. The relationship between episodic memory and context: Clues from memory errors made while under stress. *Physiological Research*, pages S3–S11, 2002.

[58] D. Newston and G. Engquist. The perceptual organization of ongoing behavior. *Journal of Experimental Psychology Social Psychology*, pages 436–450, 1976.

[59] N. Oliver, G. Smith, C. Thakkar, and A. C. Surendran. Swish: semantic analysis of window titles and switching history. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 194–201, New York, NY, USA, 2006. ACM.

[60] L. Perlow. The Time Famine: Toward a Sociology of Work Time. *Administrative Science Quarterly*, 44(1):57–59, 1999.

[61] A. Raskin. Feature the linguistic command line. *interactions*, 15(1):19–22, 2008.

[62] T. Rattenbury and J. Canny. Caad: an automatic task support system. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 687–696, New York, NY, USA, 2007. ACM.

[63] G. Robertson, E. Horvitz, M. Czerwinski, P. Baudisch, D. R. Hutchings, B. Meyers, D. Robbins, and G. Smith. Scalable fabric: flexible task management. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 85–89, New York, NY, USA, 2004. ACM.

[64] J. Robinson. *Autobiographical memory*, chapter Temporal reference systems and autobiographical memory, pages 159–188. Cambridge University Press.

[65] M. Rouncefield, J. Hughes, T. Rodden, and S. Viller. Working with constant interruption: CSCW and the small office. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, page 286. ACM, 1994.

[66] K. Ruhleder and B. Jordan. *Information Systems and Qualitative Research*, chapter Capturing Complex, Distributed Activities: Video-Based Interaction Analysis as a Component of Workplace Ethnography, pages 246–275. Chapman and Hall, London, 1997.

[67] S. Schwan and B. Garsoffky. The cognitive representation of filmic event summaries. *Applied Cognitive Psychology*, pages 37–56, 2004.

[68] A. J. Sellen, A. Fogg, M. Aitken, S. Hodges, C. Rother, and K. Wood. Do life-logging technologies support memory for the past?: an experimental study using sensecam. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90, New York, NY, USA, 2007. ACM.

[69] A. J. Sellen and S. Whittaker. Beyond total capture: a constructive critique of lifelogging. *Commun. ACM*, 53:70–77, May 2010.

[70] M. Shum. The role of temporal landmarks in autobiographical memory processes. *Psychological Bulletin*, 124(3):423–442, 1998.

[71] C. S. Smith. *Modes of discourse : the local structure of texts*. Cambridge University Press, 2003.

[72] L. Sproull. The nature of managerial attention. *Advances in information processing in organizations*, 1:9–27, 1984.

[73] A. Strauss. Work and the Division of Labor. *Sociological Quarterly*, 26(1):1–19, 1985.

[74] N. M. Su and G. Mark. Communication chains and multitasking. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 83–92, New York, NY, USA, 2008. ACM.

[75] J. C. Tang, S. B. Liu, M. Muller, J. Lin, and C. Drews. Unobtrusive but invasive: using screen recording to collect field data on computer-mediated interaction. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 479–482, New York, NY, USA, 2006. ACM.

[76] C. Tashman. Windowscape: a task oriented window manager. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 77–80, New York, NY, USA, 2006. ACM.

[77] E. Tulving. Episodic memory: from mind to brain. *Annual Review of Psychology*, 53:1–25, 2002.

[78] B. Tversky, J. M. Zacks, and B. Martin. *Understanding events: How humans see, represent, and act on events*, chapter The structure of experience. Oxford University Press.

[79] G. Wang and W. Zhang. A steganalysis-based approach to comprehensive identification and characterization of functional regulatory elements. *Genome Biology*, 7:R49+, June 2006.

[80] A. Welford. Single-channel operation in the brain. *Acta Psychologica*, 27:5–22, 1967.

[81] S. Whittaker, D. Frohlich, and O. Daly-Jones. Informal workplace communication: what is it like and how might we support it? In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, page 137. ACM, 1994.

[82] J. M. Zacks, N. K. Speer, K. M. Swallow, T. S. Braver, and J. R. Reynolds. Event perception: A mind/brain perspective. *Psychological Bulletin*, pages 273–293.