AN ABSTRACT OF THE THESIS OF

Seyed Soroush Ghorashi for the degree of Master of Science in Computer Science presented on December 7, 2011

Title: Leyline, a Provenance-based Desktop Search System Using a Graphical Sketchpad Interface.

Abstract approved:

Carlos Jensen

While there are powerful keyword search systems that index all kinds of resources including emails and web pages, people have trouble recalling semantic facts such as the name, location, edit dates and keywords that uniquely identifies resources in their personal repositories. Reusing information exasperates this problem. A rarely used approach is to leverage episodic memory of file provenance. Provenance is traditionally defined as "the history of ownership of a valued object". In terms of documents, we consider not only the ownership, but also the operations performed on the document, especially those that related it to other people, events, or resources. This thesis investigates the potential advantages of using provenance data in desktop search, and consists of two manuscripts. First, a numerical analysis using field data from a longitudinal study shows that provenance information can effectively be used to identify files and resources in realistic repositories. We introduce the Leyline, the first provenance-based search system that supports dynamic relations between files and resources such as copy/paste, save as, file rename. The Leyline allows users to search by drawing search queries as graphs in a sketchpad. The Leyline overlays provenance information that may help users identify targets or explore information flow. A limited controlled experiment showed that this approach is feasible in terms of time and effort. Second, we explore the design of the Leyline, compare it to previous

provenance-based desktop search systems, including their underlying assumptions and focus, search coverage and flexibility, and features and limitations.

Leyline: a Provenance-based Desktop Search System Using a Graphical Sketchpad Interface

by
Seyed Soroush Ghorashi

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented December 7, 2011
Commencement June 2012

Master of Science thesis of Seyed Soroush Ghorashi presented on December 7, 2011

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Seyed Soroush Ghorashi, Author

ACKNOWLEDGEMENTS

First and foremost I want to thank my Major Adviser, Dr. Carlos Jensen for all his guidance and support during this thesis work. I couldn’t be here without your guidance and support.

I would also like to thank my parents Abolhasan, Sorour, and my sister Sally and my brother Soheil, for their endless encouragement throughout my study. I feel lucky to have such an amazing and supportive family behind me.

I thank all my friends at OSU and HCI research group members for their support throughout my graduate school life.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table Page

1. INTRODUCTION

The increase of cheap and abundant local storage on computers and other devices, as well as the growing trend of cloud storage and computing makes information management and retrieval one of the most challenging tasks facing computer users. These trends drive a significant increase in the amount of information stored and available, making it difficult to find and retrieve specific file or resources. Users' best mechanism for finding and retrieving files is to organize them carefully in nested folders for later manual searches (Barreau & Nardi 1995) (Bergman et al. 2008) (Teevan et al. 2004). However, current trends in pricing, storage density and the proliferation of cloud drives, discourage users from managing their information due to an abundance of space and resources, and by requiring users to devote more time and energy to do this kind of organization and cleanup. Therefore, users are increasingly reliant on keyword search tools.

While these tools have become proficient at indexing all types of resources, including emails and web history, recalling a set of keywords or meta-data that uniquely identify a resource becomes challenging. Considering information reuse and storing different copies and drafts of files in different locations, this problem is only exasperated. As a result, keyword search often returns a large number of hits that requires user to manually scan through and identify the desired file among a potentially large number of possible candidates.

Some have done research on augmenting traditional keyword search using provenance data. Provenance generally means "the history of ownership of a valued object" (Merriam-Webster Online Dictionary), but in terms of documents, we also consider the operations performed on the document, especially those that relate it to other documents, people or events. Provenance data differs from other kinds of meta-data because it is based on relation between resources and adding said data also allows us to leverage episodic memory to perform fundamentally different searches.

This thesis manuscript consists of two papers. In the first paper, through a mathematical analysis of the data from a longitudinal user study, we show that provenance data can be used to narrow search space effectively. This analysis also shows us the expected complexity of queries. Based on this, we designed an intuitive graphical sketchpad interface for search query composition. This interface is based on provenance graphs that have been proved to be effective memory cues (Jensen et al. 2010).

We then introduce Leyline. The Leyline is the first search system that uses provenance graphs both in query composition and results presentation. We evaluated the Leyline through a limited lab experiment where results show that our UI approach is fast, effective and easy to learn.

In the second paper, we discuss Leyline's design and architecture and describe its different parts in more details:

- A user interface (UI) that allows users to compose queries as directed graphs representing provenance events, demonstrated to be effective cues in recall.
- Methods for monitoring provenance events, algorithms to efficiently search using this metadata, and let users manipulate results to aid recall.

Afterwards we examine the most successful and recent provenance-based search systems, analyze their relative strength and weaknesses in these following areas:

- What kind of provenance data they use.
- Where they use provenance data.
- How they gather provenance data.
- Their UI approach for composing query.
- How they present results.
- How these tools were evaluated.

Finally, through a comparative analysis we show how Leyline's design and features address the shortcomings of other systems.

2 Provenance-Based Search Using a Graphical Sketchpad

Soroush Ghorashi, Carlos Jensen

School of EECS
Oregon State University
Corvallis, Oregon, 97331, USA
{ghorashi, cjensen}@eecs.oregonstate.edu

Proceedings of the 2012 annual conference on Human factors in computing systems
Austin, TX, USA
(Under evaluation)

2.1 ABSTRACT

People have trouble correctly recalling semantic facts such as the name, location, edit dates, and keywords of files in their repositories. These are the mechanisms, quite sensitive to mistakes, typically available for search. A rarely used approach is to leverage episodic memory of file provenance; the creation, use and sharing of documents, such as whom the file was emailed to, or what websites were used as sources. The addition of this data could result in more information with which to search. To investigate the possible advantages of using provenance data in search, we developed a prototype system based on what has been learned from a longitudinal study of how knowledge workers create, use, and search for information. In this paper we show that provenance data is useful and popular in search, and that an interface based on a graphical sketchpad is technically and logistically feasible.

2.2 INTRODUCTION

The availability of abundant and cheap storage makes every computer a potential massive repository of documents, spreadsheets, presentations, emails, pictures, music and videos. If current trends in pricing and storage density continue, users aren't likely to devote the time and energy required to carefully organize their files and information within current file storage models. The same models, with non-overlapping folders as project delimiters, poorly match the way we work; we frequently reuse information and files from project to project. Current trends in cloud storage are only expected to exasperate the problem, with local and remote copies of files, and increased difficulties with search.

Significant increases in the amount of information stored and processed by information workers make it difficult to find and retrieve a specific file or resource in their repositories. Users' best mechanism for finding files is to carefully arrange these in nested folders and later perform manual searches (Barreau & Nardi 1995) (Bergman et al. 2008) (Teevan et al. 2004), but this model breaks down as the number of nested folders increases, as we distribute content across locations, and collaborate with people with different organizational schemes (often idiosyncratic) (Rader 2010). Instead, users have to increasingly rely on keyword search.

While keyword search systems have become proficient at indexing all forms of resources, including emails and web pages visited, recalling a set of keywords that uniquely identifies a resource becomes challenging. This is only exasperated as we reuse information and as we are encouraged to store multiple copies and drafts of files. As a result, keyword search often result in a large number of hits, and the user has to manually identify the correct target. Some have examined the use of provenance data to enhance traditional keyword search. Provenance has been defined as “the history of ownership of a valued object” (Merriam-Webster Online Dictionary). Using provenance data in the search process not only means that there is more data to potentially uniquely identify files and resources, but also that we can leverage episodic memory.

Provenance data is attractive because while semantic facts are easily forgotten or confused over time, episodic memory has a tendency to remain more cohesive (Tulving & Thomson 1973). Furthermore, it enables weakly linked teams or groups with the ability to conduct successful searches by utilizing what they know about the workflow. For instance, a user may not remember a file-name accurately, or a sufficiently unique document keyword, but they may remember emailing a specific colleague about the document and use this to narrow the list of potential documents.

Search is an important problem for knowledge workers, who make up to 43% of the U.S. workforce (Bureau of Labor Statistics). Knowledge work is a term used to encapsulate a broad set of jobs and activities, including IT, management, analysis, accounting, marketing, design, law, etc. Today’s knowledge workers are increasingly multi-tasking. In (Chudoba et al. 2005), two-thirds of employees worked on between three and five teams. Knowledge workers deal with a large volume of email and phone communication (Jensen et al. 2010), and keep large repositories of documents, links and data. Managing these requires overhead and few users spend the time (Boardman

& Sasse 2004). Such maintenance is primarily performed at milestones such as end of projects or major deliverables, and is instrumental to the success of search and reuse (Ravasio & Schär & Krueger 2004).

Reuse is an important strategy; old documents are used as templates or as sources. No matter how information is reused, it is important that reuse is tracked in order to facilitate auditing and prevent intentional or unintentional plagiarism. Doing so requires us to rethink the way our file systems are organized, and the tools to track information and make such information available to users.

Research has shown that provenance information is plentiful, meaningful, and easily recalled by users (Jensen et al. 2010). Since relationships are more difficult to express than keywords, it begs the question of how best to use provenance information to augment operations like search. In this paper we examine how efficient query composition can be for provenance search, as well as the results of a limited evaluation of a graphical sketchpad interface to file search using provenance information on a large file repository.

The rest of this paper starts off with a description of the related work. We then address our two research questions separately; first a description of our methodology and results with regard to the feasibility of using provenance to narrow search, and then a description of the methodology and results of our specific approach to doing such a search. We conclude with a discussion of the remaining challenges in the design of provenance-based search tools.

2.3 RELATED WORK

Knowledge workers add value through the information they produce, and their ability to apply it to develop new understanding (Davis 2002) (Drucker & Peter 2009) (Kidd et al. 1994). Therefore, a growing body of work has focused on helping them deal with growing file repositories. A number of commercial tools (e.g. Google Desktop, Windows Desktop Search, OS X Spotlight) have been created to help users find information, primarily using keywords. However, as repositories grow, these tools become less effective due to a lack of unique and memorable keywords. Users therefore have to sort through lists of potential candidates. This is inevitable as most knowledge workers are specialists working in narrow domains, and thus their files have overlapping information and terminology. This is exasperated by reuse (copy-paste), and storing drafts of documents.

Research has been done on augmenting keyword search using provenance information. Provenance is an effective trigger for other memories related to files and folders (Blanc-Brude & Scapin 2007). Because of this “packaging”, provenance may be useful in search. In a file system context, we are interested in tracking operations performed on files, especially those that tie documents to people and events.

While studies of provenance in real-world settings are rare, a study at Intel Corporation (Jensen et al. 2010) showed that provenance events are common, and that provenance can be used to weave large and complex networks of files, emails (and people), webpages, and other resources. This study found that graph representations are understandable and good representation, and that they can help users reason about their work process.

A number of systems have been developed to track and use provenance information. These fall into one of three camps; those integrated into the operating system, those monitoring applications, and those reconstructing provenance events from file meta-data. The Provenance-Aware Storage System (PASS) (Muniswamy-Reddy et al. 2006) put a provenance monitoring system into an operating system. One problem with this approach is that data is collected at the event level and requires processing to identify high-level events.

A second approach is to have an application-monitoring service. An example of such a system is TaskTracer (Dragunov et al. 2005), built to help users recover from interruption, tracking files and application events in the process. This includes operations such as move, rename, save-as, copy/paste, email attachment, etc. The problem with this approach is that individual applications have to be instrumented.

A third approach is to infer provenance based on existing meta-data, such as modification dates, directory structures, or email logs. This is the approach used by Google Desktop. The advantage of this approach is that it easily accommodates existing repositories and is relatively simple to implement, as it does not require integration with individual applications or the operating system. However, this means that a number of important provenance events are missed, including copy-paste operations, found to be the majority of provenance events (Jensen et al. 2010).

A number of tools have made this information available to users for search purposes. Feldspar (Chau & Myers & Faulring 2008) is a tool that extracts information from Google Desktop's database and generates an association graph. This graph can then be queried by users through an interface that allows users to specify relationships using a flow-chart like model.

Quill is a tool that adopts a narrative model; presenting users with template story statement and allowing them to fill in the blanks about the target document (Gonçalves & Jorge 2008). Although it tracks meta-data about documents, email attachments, web pages, and calendars, it does not store dynamic relations between files such as copy/paste, save-as and other file system operations.

YouPivot is a tool that allows users to search through their computer history for memorable context. It allows users to mark a moment in time as being important. Users can later browse through their computer history around these key moments in time, using these markers as beacons (Hailpern et al. 2011).

There are other tools that use traditional keyword-based input to identify possible files, but use provenance to reorder or expand results, such as Connections (Soules & Ganger 2005). Another tool is Beagle++, which uses email, web page, and document meta-data to order search result based on contextual information, which is similar to provenance (Chirita et al. 2006).

2.4 METHODOLOGY

Although provenance tools have shown promise, important questions remain; including the crucial "how well do these interfaces scale as repositories grows?" These tools have not always been tested using representative data, or used common provenance events. This is important, because if a relationship is too common or forgettable it may not help the search. However, if an event is too rare, it may not be worth including in an interface. Furthermore, while provenance is commonly shown as a directed graph (Figure 2.1), something users like, few if any search systems have tried to use it as a UI model for composing a query.

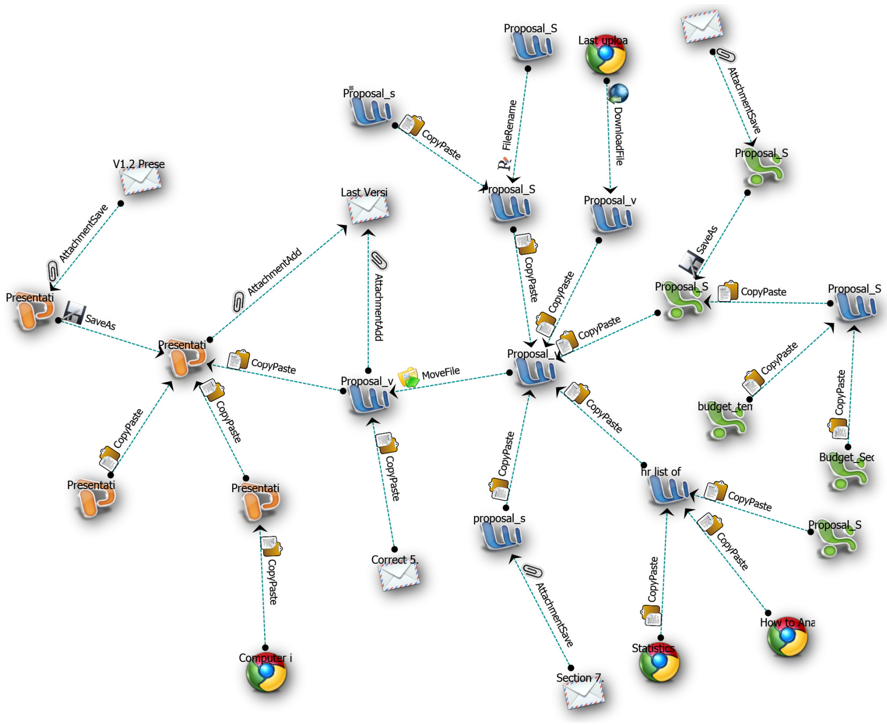
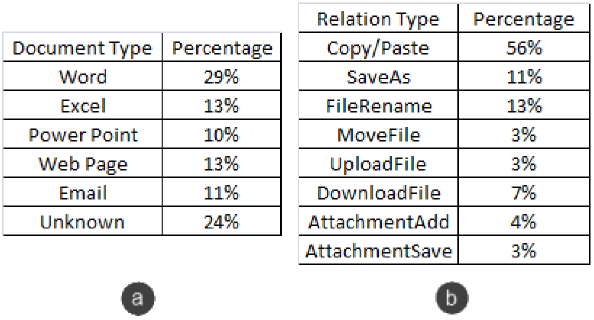


Figure 2.1 Sample provenance graph.

Our goal was to determine whether a graphical sketchpad is feasible for query composition, and evaluate this approach using realistic data. We will discuss the methodology and results for the first research question here, and the methodology and results for the second part of the research questions later.

Table 2.1 (a) Documents by type. (b) Provenance relations by type from Jensen et al. 2010.



Document Type	Percentage
Word	29%
Excel	13%
Power Point	10%
Web Page	13%
Email	11%
Unknown	24%

Relation Type	Percentage
Copy/Paste	56%
SaveAs	11%
FileRename	13%
MoveFile	3%
UploadFile	3%
DownloadFile	7%
AttachmentAdd	4%
AttachmentSave	3%

In order to do the first evaluation, we leveraged the data gathered from an existing longitudinal study at Intel Corporation, as described in (Jensen et al. 2010). In this study, 17 knowledge workers from Intel Corporation were monitored for over two months using Tasktracer (Dragunov et al. 2005). Summary statistics of their findings are shown in Table 2.1. While users had much larger file repositories (in terms of number of documents) than are shown in Table 2.2, these were not in the active set over their 3-month study, and may be less important for search.

Table 2.2 Overview of data-distribution from Jensen et al. 2010.

Participant ID	# of Documents	# of Relations	# of Sub-graphs	Unique walks w/o Relation Type	Unique walks w Relation Type	Max Length of unique walks w/o Relations	Max Length of unique walks w Relations
1	108	93	31	40	40	3	3
2	1,003	875	293	320	328	7	6
3	418	414	107	138	141	7	7
4	299	283	81	99	99	8	4
5	510	470	160	186	186	4	4
6	44	36	16	19	19	4	3
7	87	107	28	32	32	3	3
8	935	708	362	386	392	5	4
9	150	150	59	63	63	4	3
10	140	162	40	48	48	4	4
11	239	184	88	99	99	4	4
12	172	154	52	62	63	6	6
13	273	405	58	80	82	7	5
14	369	357	120	140	140	4	4
15	291	334	88	103	105	5	5
16	518	682	120	149	157	10	9
Mean	347.25	338.38	106.43	122.75	124.63	5.31	4.63
Median	282	308.5	84.5	99	99	4.5	4
Std. Dev.	281.52	245.38	95.46	101.80	103.94	1.99	1.67

To determine the feasibility of this query approach, we did an analysis of the graphs found in (Jensen et al. 2010). Table 2.2 has data on the worst-case for each data-point in our sample.

2.5 RESULTS

Our first goal was to determine how complex a query is needed to find a unique file, assuming less than perfect recall. Many provenance links are relatively common. For instance, copy/pasting between two documents or versioning occurs frequently. Using these events can result in long queries (see Figure 2.2). When composing queries using a graphical user interface, this becomes an issue. We therefore need to find the worst-case scenario for how many files a user has to place in a sketchpad to identify a file.

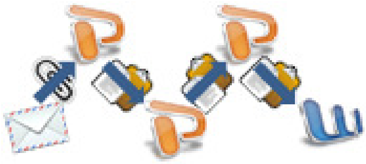


Figure 2.2 A sample chain of documents including provenance relationships between them.

We found all unique walks in the provenance graphs from (Jensen et al. 2010), generating strings with and without consideration for the type of link between the documents. Next we found repeating sub-strings within these walks. These represent a worst-case scenario for search because they represent repeating patterns, and thus where we will need additional work to disambiguate a search result. We therefore found the longest repeating string for each participant. In the worst case, a user would need to draw a graph one greater than the longest repeating string in order to guarantee a unique search result (or there is no unique sequence).

As we see from Table 2.2, this maximum length ranged from 3 to 10 (median 4.5) if we ignore relationship types, or 3 to 9 if we include these (median 4). This is important because it tells us that even if people do not remember the relationship type, we lose relatively little search power. The reason for this is because 80% of all provenance links are copy-paste and versioning, especially in long chains.

Working style has a very large effect on these numbers, more so than the total size of the file repository. Some users reuse via copy-paste and engage in more versioning, others choose to retype and reuse that way, and work on the same file from start to finish. As we can see in Figure 2.3, the maximum walk distance, or the worst case for provenance search, appears to grow linearly with the number of provenance links, at a rate of approximately 1 more node per 200 links.

Whether this scaling will hold as repositories grow over prolonged periods of time is unknown, but it is not necessarily unreasonable to assume it will. Most reuse is relatively temporally constrained, and though we might develop longer, skinnier, loosely connected graphs, these do not add as much complexity to search as bushy, highly connected graphs do. This is however something that will have to be investigated more thoroughly in the future.

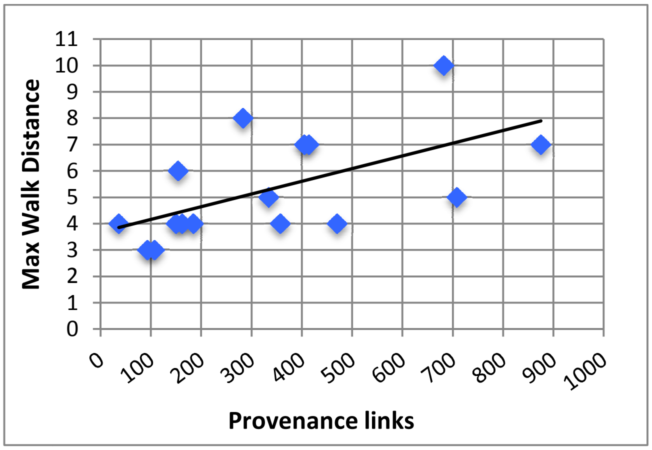


Figure 2.3 Provenance links vs. Max Walk Distance.

It therefore seems that provenance can be used to rapidly narrow the search, and that the number of nodes needed in a query would on average be relatively small (median 4). Furthermore, even vague recollection (not remembering the type of link) does little to increase the complexity.

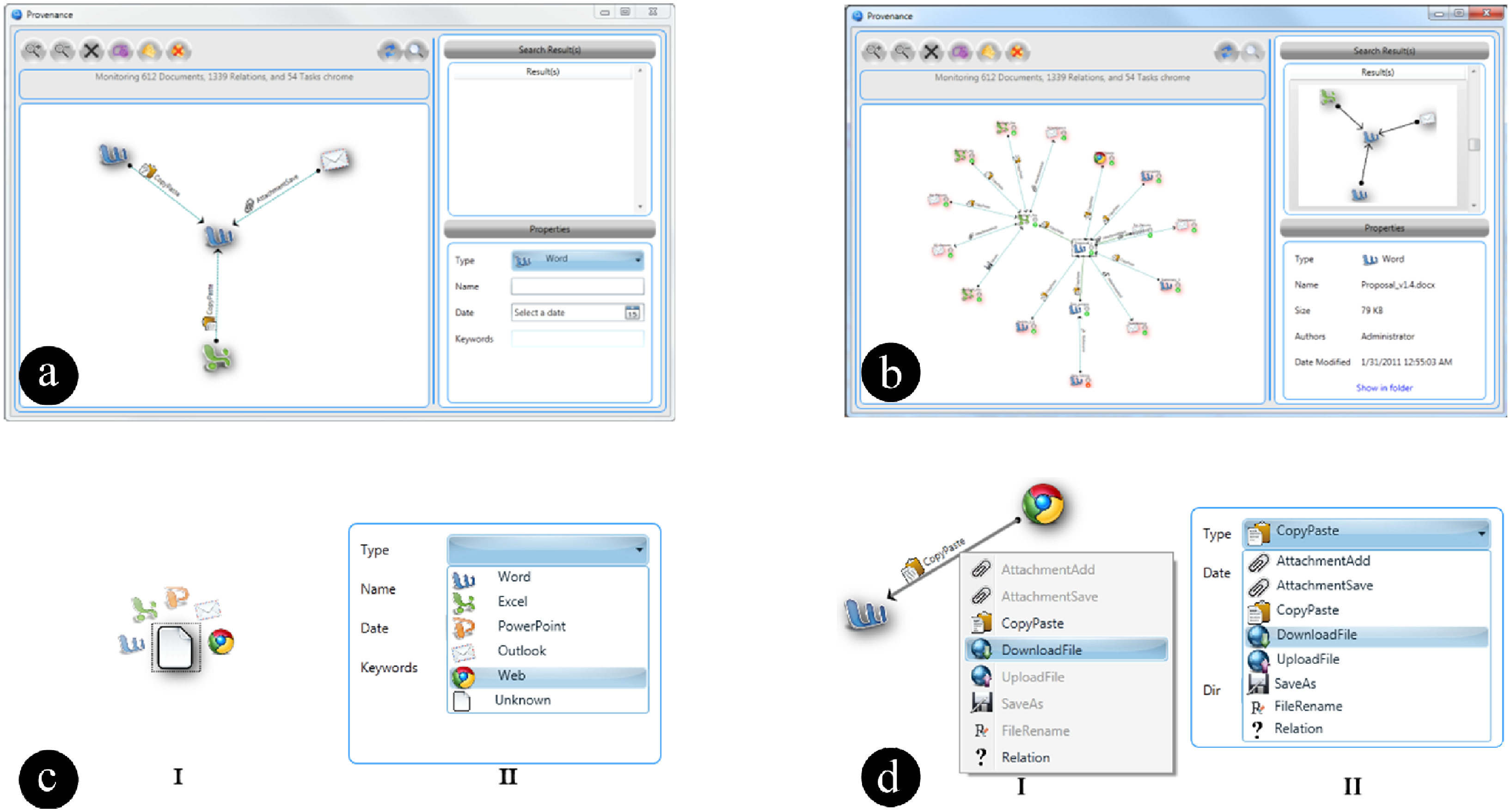
2.6 DESIGNING A PROVENANCE –BASED SEARCH SYSTEM

An important aspect of human cognition is that recognition is easier than recall (Bates 1998). According to Jensen et al. (2010) provenance graphs aided recall about documents, tasks and workflow. There was a good match between the provenance events users recalled, and the most common. Given the success of this representation, we designed our interface around this concept in the hope that it would aid users in reasoning and recalling context.

Using the Jensen et al. (Jensen et al. 2010) study as a guide, we built a prototype around the idea of a direct manipulation graphical sketchpad for query construction and result display. Provenance is shown as a directed graph, where nodes are documents, emails, web pages and other artifacts, and the links between them are provenance events such as accessing, copy-pasting, versioning, emailing, etc.

To search, a user draws a graph in a canvas (see Figure 2.4a). Interactions with the query area are kept simple to minimize the need for training and keep the interface efficient. A user can add a document to a query by clicking on a blank area of canvas and a menu will appear to let them set the document type (optional). If the user wants to change the document type, they can right click on the document icon to get a context menu (Figure 2.4c.I), or they can use the properties panel on the right of application window (Figure 2.4c.II). Our system currently supports six resource types: Word, Excel, PowerPoint, Web, Email, and a catch-all “Unknown”.

A provenance link is established between two resources by clicking on one (source) and dragging the pointer to another (target) while holding down the mouse-button. Directionality is determined by the direction of movement, but can be changed through the properties panel. The type of relationship can be set by either clicking on a link and selecting from a context menu (see Figure 2.4d.I) or through the properties panel (Figure 2.4d.II). Provenance relationships supported includes: Attachment_Add, Attachment_Save, Copy_Paste, Download_File, File_Rename, Upload_File and the catch-all “Unknown” relation. Only relationships that make sense are available. For instance, a web page can be saved, downloaded, and copy/pasted from, but it cannot be renamed. This simplifies things for users, and makes sure the resulting queries are more successful.



Type: Word
Name:
Date: Select a date
Keywords:

Type: Word
Name: Proposal_v1.4.docx
Size: 79 KB
Authors: Administrator
Date Modified: 1/31/2011 12:55:03 AM

Figure 0.4 Figure 2.4 (a) A sample query, looking for a word document saved from an attachment and with information pasted to and from two other documents. (b) Expanded result. (c) Menu to set document type. (d) Menu to set relation type.

The goal of using provenance is to enhance search, and not to exclude the use of other successful strategies, like keyword search. Therefore, users can add keywords, partial or complete file names, and/or dates to any file in the graph through the properties panel (see Figure 2.4a). This gives users more flexibility in terms of leveraging what they recall about a document. Finally, the preference panel allows the user to specify which document in a graph they are actually searching for, as the target can often be a peripheral part of the graph. This is not a required part of a query; a user can use graphs to specify a starting point for exploration, expanding the networks of use (see Figure 2.4b).

Given a query graph we can proceed to search our database, trying to find a pattern match. For each query we have to solve a sub-graph isomorphism problem, which is known to be an np-complete problem. However, we know that user recall is often imperfect, and thus it is desirable for search queries to be flexible, identifying close misses. In Figure 2.5 we see an example of this, where a user remembers that there is a relationship between two documents, but not necessarily that there were intermediate steps, with another document acting as an intermediary. We refer to these as "star relations" in that any relation link could in theory be replaced by an arbitrary set of nodes and relationships. Adding this "flexibility" to the system also changes the complexity of the search algorithm in our favor.

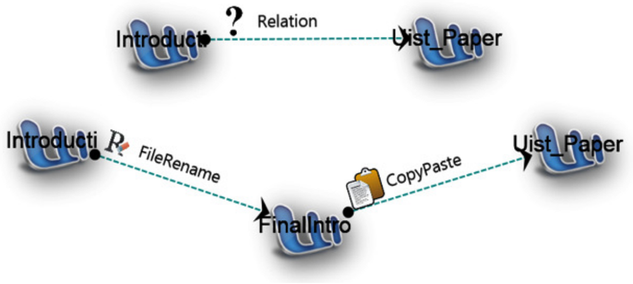


Figure 2.5 Star links.

By allowing flexible graph mutations, we can use the G-Ray algorithm (Tong et al. 2007), which is fairly fast, scalable, and does approximate matching in addition to exact pattern matching. This algorithm works with big attributed graphs whose nodes have one categorical type. Before finding matched patterns using G-Ray we have to do some substitutions as both our nodes and edges have categorical attributes, not supported by G-Ray. We therefore substitute each original edge with two attribute free edges tied to a dummy node with an edge type.

Now that we have our query, we can run it against our provenance graphs and use G-Ray's goodness function to rank results. If keywords, dates, or full or partial file names were provided, we run candidate graph nodes through a keyword search system to filter potential matches.

We present potential matches as thumbnail graphs in the result panel (top right, Figure 2.4b). Selecting one of these candidates expands it in the main canvas, and files can be explored, with additional information given as tooltips. User can select or hover over a file or provenance link to see when the event took place, where a file is stored, when it was created, or when it was last used. This semantic information is often highly meaningful for triggering or reinforcing episodic memory.

In addition to the semantic cues, we overlay provenance information that may help users identify targets or explore information flow (see Figure 2.6). Because provenance tracks the history of documents, it is not only possible, but likely for search queries to return references to deleted files. To indicate the status of a file or resource, we use a colored indicator (green bubble in Figure 2.6). If a resource is available, we draw a green bubble below and to the right of the resource icon. If the resource is no longer available, the dot is red. Double clicking on a resource with a green indicator opens the resource, be it a file, a webpage, or an email. Clicking on a deleted resource does not do anything.

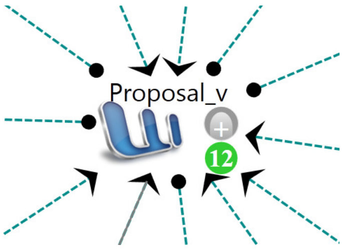


Figure 2.6 Magnified view of a node in result graph.

We know from previous studies that provenance graphs are effective memory cues. However, graphs can be large, and this makes it sometimes difficult to get an overview of the graph, or to focus users' attention to key areas of the graph. There is therefore a strong need to "limit" the complexity of provenance graphs in search results, while still allowing for enough flexibility to allow for exploration.

In our prototype we chose to limit complexity by collapsing, or hiding nodes which were not part of the sub-graph the G-Ray algorithm used to match the result. To indicate to users that more links and files exist, we overload the availability indicator by writing the number of directly connected nodes currently not being displayed in the graph. If there are hidden nodes, we place a gray bubble over the availability indicator (with a + or – sign) as a toggle for user to expand or collapse the next level of a sub-graph. The nodes in the expanded sub-graph will have the same indicators available, and can be further expanded. An example of this expansion can be seen in Figure 2.4b, where the main window shows an expanded graph, and the top right corner shows the original query result.

While we think the design decisions we have made with regards to availability indicators and how to constrain the complexity of search results will help users in their search and exploration tasks, we don't know that this is necessarily so, or that our design is the most optimal or intuitive. This requires more careful future study.

2.7 PROTOTYPE EVALUATION

The goal of this part of our study was to examine whether our graphical query composition approach was feasible in terms of time and effort. Search is used frequently, so this process needs to be efficient and easy to use. If a graphical approach to query composition takes too long, or requires too much thought, it will fail to see widespread adoption. At the same time, the advantage of using provenance is that it allows users to perform fundamentally different searches. A direct comparison to keyword search is difficult, as is setting an acceptability threshold for query composition. We set an arbitrary threshold at the two-minute mark (query composition, search, visual inspection, and target identification), and collected both qualitative and quantitative measures to evaluate the approach.

To evaluate our prototype, we created a file repository modeled after those seen by Jensen et al. (2010). This data-set modeled a small group working on a set of reports, emails and spreadsheets. We did this in order to avoid using the original files, as these contain sensitive information. Our sample was made to be close to a worst-case scenario within the parameters of the data in Table 2.2. A summary of this sample can be found in Table 2.3.

Table 2.3 Overview of sample created for study part 2.

Number of Documents	612
Number of Relations	1,339
Number of Sub-graphs	53
Unique walks w/o Relation Type	99
Unique walks w Relation Type	112
Max Length of unique walks w/o Relations	9
Max Length of unique walks w Relations	8

2.7.1 Participants

We recruited volunteers from the graduate student body of a large university from disciplines that would qualify as knowledge workers. We chose to work with graduate students instead of undergraduates because we assumed graduates would have more experience and a greater need to manage knowledge repositories. Participants were asked to complete a short screening online questionnaire to help us understand their experience with file search and search tools. We selected ten volunteers who had extensive knowledge of search techniques and tools, and/or about information reuse in their own work. Participants' ages ranged from 23 to 35 with an average of 27. Half of our participants were female and no participant had prior experience with our prototype. Each session lasted for 60 minutes, including training, and participants were given$10 as compensation for their time.

2.7.2 Experiment Setup

The scenario presented to participants, and which drove the creation of the data repository was the following: A university team is working on a proposal, with different drafts and sub-documents (textual descriptions, diagrams, etc.) being developed by different team members and shared through a shared directory and email. In parallel, a group of accountants are working on a budget analysis to go along with the proposal. In order to make this process work the two groups have to not only exchange information and drafts within their groups, but between them as well; the technical team needs to communicate scope and needs to the accountants, and the accountants need to communicate constraints and resource availability to the technical team. If everything works as it should, these two teams will create a set of documents, including a final proposal word document, a final budget in an excel sheet, and a PowerPoint presentation as a summary to a manager.

Because participants were not part of the repository creation process (creating the repository took weeks), we could not test their ability to recall keywords, file properties, or anything about the creation process. We instead asked them to assume the role of a newcomer to the process needing to learn about or fix mistakes made as part of this process. Users were given 9 search tasks to perform, in random order, as explained in Table 2.4.

Table 2.4 Experimental tasks.

1	Find the email from *Dedrie* about (**keyword**: *presentation introduction)* which contains two files for the introduction.
2	The (**file name**: *budget_Section3.5*) excel document, contains an embarrassing mistake. Who might have made that mistake and who else might be affected?
3	(**File name:** *Proposal_v4.5* word) with two other files was sent to someone. Who did you send them to? What were those files names and where are they?
4	List the word documents you've downloaded from the Internet which are still available on the hard drive.
5	You used a word and an excel document to create a (**keyword**: *presentation_costdocument*) word document. You used this in the first version of a PowerPoint presentation. Who did you send this to?
6	Find the word document you created using information copy/pasted from an email, a web page, and an excel document. Find the emails that have this word document as an attachment.
7	You saved a word document from an email attachment. Together with an existing excel document, you created a word document called (**file name**: *proposal*). You used this document to make a presentation. Who sent you the first word document?
8	You used a word file to create an excel document, and then you used that in another excel document. What is the name of second excel document? (Expand the result to see the process)
9	You downloaded a word document from Internet and used this along with an excel document and another word document to create and complete section 14 of the proposal (**keyword**: *proposal section 14*) in a separate word document. What site you downloaded the word document from and in which version of proposal document did you use section 14?

We modeled these tasks after the descriptions participants gave when discussing their workflow and information reuse in the study performed by Jensen et al. (2010). Before the study, we conducted a pilot to determine that the questions were understandable and matched how a naïve user would express relationships and workflows informally. We also verified that a file or resource could be found with the information provided, and that there was more than one search result for at least half the searches even if performed optimally (including at least one instance where the correct answer target would be guaranteed to not be the top ranked result).If participants pursued a sub-optimal query strategy, they would potentially be presented with more results (more ambiguity), or suboptimal result ranking. We wanted to make sure that all participants would have to deal with some ambiguity, but we did not control for whether optimal queries were formulated, given that participants had to rely on our description of the search terms.

We encouraged participants to think aloud during the experiment so we could follow their thought process. We also captured their screen for more usability analysis. Participants were told that they have four minutes for each task and they are not allowed to move on to next task until either the current task is done or the four minutes had passed. If they used more than two minutes, we marked the trial as a failure, but allowed participants to continue for another two minutes in order to determine if they would eventually find the answer, or if they would give up.

Participants would call out when they thought they had found the correct answer. The experimenter would confirm whether they had performed the task correctly, and if they had arrived at an incorrect answer while still having time available they were instructed to keep trying until they either found the correct answer or ran out of time.

After finishing the nine search tasks, we interviewed participants about their experience with our prototype. We asked general questions about their overall experience with the user interface design and concepts, features, and whether they would like to have such a tool available on their personal computers.

2.8 RESULTS OF PROTOTYPE EVALUATION

We only had one participant fail at one task (task 2, timed out at 4 minutes, which accounts for the high std. dev. for this task). Table 2.5 gives an overview of the average time needed to compose each query. It is important to keep in mind that these times included the time needed to figure out the task and evaluate alternative solutions, not just query composition. We had a mix of simple and harder tasks. For the simple tasks, the average completion time ranged from between 72 and 93 seconds, within our success metric and harder tasks with an average completion time between 126 and 155 seconds, over our success metric. For the trials that were successfully completed (all but one), the slowest completion time was 173 seconds and fastest 53.

Table 2.5 Experimental results.

Q	# Successes (≤2 minutes)	Query complexity		Avg. completion time (seconds)	Std. Dev.
		# Nodes	# Links		
1	10	1	0	73.3	8.001
2	0	1.2	0.5	155.4	33.778
3	10	1.3	0.3	76.8	8.324
4	10	2	1	71.7	11.548
5	2	3.4	3	126.1	10.796
6	10	3	3	78.8	10.475
7	0	4.8	3.8	135.8	12.796
8	10	3	2	92.7	10.874
9	0	5.4	4.4	144.0	9.381
Avg.	**57.8%**	**2.79**	**2.00**	**106.1**	

As we can see from the standard deviations, the completion times within each condition were very consistent. We see this as an encouraging sign, as it means that our interface and the general approach was understandable and usable, no one got lost or confused. When we plot the query complexity (in terms of number of nodes and links in the query graph, see Figure 2.7) we see that it aligns reasonably well with completion time, but does not grow aggressively. This is a positive indicator in terms of potential scalability.

Based on our observations, we found that participants read the complete task description before stating, identified the target resource in the description, and then built the query graph out from this target in order of appearance in the task description. We also found that users worked on one resource or relation at a time, providing all details at once rather than building a rough picture and then going back and adding additional details.

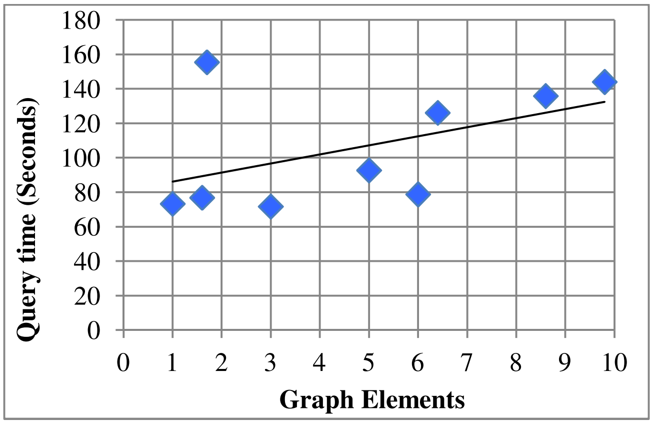


Figure 2.7 Completion time vs. Query complexity.

After the task, we asked participants to rate their satisfaction with our user interface using a likert-scale where 1 meant very dissatisfied and 5 meant very satisfied. The average satisfaction rating was 4.2 ($\sigma = 0.4$). Nine out of the ten participants indicated that they would like to have access to a search tool such as ours on their personal computers and especially in the work environment;

> *"I would like to use this. I can see this being very helpful in [my] work environment."*

> *"Being able to search through and find associated files of different types would be great for use with school or other collaborative projects."*

Participants generally liked the interface and our approach to query composition:

> *"I liked the way to create the query because it was very visual as opposed to using the solely text-based keyword searches."*

> *"The icons for different types of documents are well designed which makes it easy to distinguish between different documents. [Automatic filtering] in setting relation type saves time and minimizes making mistakes in setting the query."*

All participants preferred the context menu to the properties panel, regardless of whether they were composing, reviewing, or examining results (tool tip). Participants liked the expand/retract option in the graphs, the availability indicator, and being able to jump into the containing folder or opening the resource directly from the interface.

> *"I thought the results clearly showed the associations between files. Opening the various files/sites from the icons made it easy to locate and view files. Being able to see the flow of how a document was created or passed around can be very useful, especially in a collaborative environment."*

That said, participants did point out areas for improvement, though they did not always agree with each other. One example of this is in the presentation of query results:

> *"File names are disturbing. I would love, if I can see the file name only when I move my cursor over it."*

> *"[It] needs full text of file names visible. Like the hover over for more info."*

This suggests that providing customization options to accommodate user preferences and needs is important. Participants found switching between query editing and result browsing to be difficult. Three participants proposed that streamlining that process should be our top priority. Others were interested in the exploration elements of our system, and suggested we add an option to animate or manipulate the time variable in results graphs (for instance showing the provenance graph at a specific point in time):

> *"Add a 'History' box. This box will list the history of users' activities step by step when they form the search query. This box would also allow the user to click on a particular item in history and the resulting searching graph would be modified on how it was on that particular instant."*

Continuing with the exploration theme, participants suggested we track and show their query history to allow them to toggle back and more easily reuse queries. This is a use of the system that we want to support, but that we did not emphasize in this prototype. We therefore see it as an encouraging sign for our future work.

2.9 DISCUSSION

Our numerical analysis of existing provenance data to determine the expected complexity of graphical queries showed us that the graphical query composition approach should be, at least in theory, feasible in terms of complexity. Though the data available for analysis was limited to that tracked over 3 months, the complexity growth rate seems acceptable to us. This was an important for us before we could justify building our prototype.

This theoretical analysis, when combined with the results from our limited user study, shows us that the amount of time and effort users need to invest in composing queries is relatively low, and appears to scale well, at least within the parameters set in this study. In the study, participants spent on average 106 seconds composing a query, and queries had an average of 2.8 nodes and 2 edges. Not terribly complicated. More importantly however, Figure 2.7 tells us that the system appears to scale well in terms of time on task vs. complexity of query.

As we already discussed, our success threshold, set at 2 minutes, was relatively arbitrary. It is difficult to make too much of the number completion times in isolation because it is difficult to determine how representative these queries are of queries users would perform on a daily basis. This is an issue when we look for advice on how to streamline and improve our interface, and will likely require a longitudinal deployment study, allowing users to discover and develop new search strategies. Based on the data we have gathered so far, as well as the feedback participants gave us, that this UI technique is efficient, well liked, and intuitive.

That said, though participants were given an interactive tutorial in how to use the system, and were encouraged to explore the prototype until they felt comfortable, we did observe a learning effect, where performance in the first two tasks was noticeably lower than in the following seven. Because we randomized the order of the questions, this should not unduly bias any one question in Table 2.5, though we should see better performance with more training.

Through this study we learned a lot about how people think about forming queries. Participants uniformly built their queries from the target out. In other words, the first node they placed on the canvas was always the resource they were searching for. This was unexpected, we expected participants to compose queries more or less as they were described in the tasks. It will be interesting to see if this holds true when subjects make up their own queries based on their own memory as opposed to working with the tasks and descriptions we give them.

Participants also worked with one resource at a time, providing all keyword and links before moving on to adding the next resource. This was also unexpected; we envisioned that at least some of the participants would first sketch a graph, then go back to refine it. Again, this may have been an artifact of the artificial task, but if it holds in real-world setting, this could allow us to dynamically update search results so users know when to stop adding details, or give feedback on how much value each new element adds. This could help train users to write better (as in more effective) queries more quickly.

Finally, our prototype was limited, and we received a lot of feedback on areas for improvement as well as ways of expanding the scope of the tool.

2.10 FUTURE WORK

The next steps for us are to incorporate the feedback and lessons learned from the participants of our study. These span the gamut from simple UI tweaks, to exploring features to encourage more exploration. Eventually though, we will need to undertake a longitudinal study, closely modeled on (Jensen et al.), where we deploy our interface with a provenance tracking system and study how people interact with the system over an extended period. The longitudinal nature of such a study is unavoidable for two reasons; first we are interested in seeing how people adapt and incorporate the system into their day-to-day practices and second, provenance-based system are only really useful once their databases become populated.

We are also interested in exploring the feasibility of providing auto-complete-like capabilities, and how these would be implemented in a graphical UI like the one we use here. One idea could be to display potential query modifications as ghosted images in the graph, and allowing users to select these rather than manual adding. This mechanism could leverage recognition rather than recall, and help users more quickly add more meaningful nodes.

For our prototype we used provenance data captured by TaskTracer. This system primarily tracks Microsoft Office and Microsoft Windows applications and events. This limits us to specific data types and provenance relations. In the future we would like to be able to support a broader set of files such as pdf, text, images, and videos. In order to do so we will need to find new and better ways of tracking provenance information. Ideally we would like our system to be tracker agnostic, able to plug into any provenance system as long as their developer opens their provenance database to outside applications, and developers publish data schemas. This will become increasingly important as more and more of these tracking functions get incorporated into the operating system layer.

2.11 CONCLUSIONS

We have shown through a numerical analysis of field data collected as part of a longitudinal study of knowledge workers that provenance relationships can effectively be used to uniquely identify documents in representative repositories. We have also shown that we can build an efficient and intuitive user interface allowing users to build queries that closely match the way that they visualize and think about provenance relationships, using provenance graphs. This allows users to leverage existing knowledge more easily and the recognition rather than recall both when composing queries as well as evaluating results.

The search algorithms used in this prototype allows for flexibility, which mimics the kind of flexibility we see in terms of user memory (concatenation of events and relationships, here referred to as star relationships), was shown to be both effective and efficient, with users having high confidence in search results and rankings. More work however needs to be done to determine both the fit of the solution as well as its scalability.

2.12 ACKNOWLEDGMENTS

We wish to thank our colleagues in the HCI group at the School of EECS, Oregon State University for their support and help in preparing the paper. We would like to thank our study participants. We'd also like to thank TaskTracer's team for giving us access to their system monitoring software.

2.13 REFERENCES

3 Barreau, D. and Nardi, B. A. 1995. Finding and reminding: file organization from the desktop. SIGCHI Bull. 27, 3 (Jul. 1995), 39-43

4 Bates, M., "Indexing and access for digital libraries and the internet: Human, database, and domain factors", JASIS, 49, pp. 1185-1205, Nov. 1998.

5 Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N., and Whittaker, S. 2008. Improved search engines and navigation preference in personal information management. ACM Trans. Inf. Syst. 26, 4 (Sep. 2008), 1-24.

6 Boardman, R. and Sasse, M. A. "Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management". In *Proceedings* of the CHI '04, Vienna, Austria. ACM, New York, NY, P.583-590, 2004.

7 Blanc-Brude, T. and Scapin, D. L. "What do people recall about their documents?: implications for desktop search tools". In *Proceedings* of the IUI, '07, Atlanta, GA. ACM, New York, NY, P.102-111, 2007.

8 Bureau of Labor Statistics. U.S. Department of Labor. "Occupational Employment and Wages." Press release. Washington, D.C. 1 May 2009.

9 Chau, D., Myers, B., and Faulring, A. "What to do when search fails: finding information by association". In *Proceedings* of CHI '08, Florence, Italy. ACM, New York, NY, P.999-1008, 2008.

10 Chirita, S. Costache, W. Nejdl, and R. Paiu. "Beagle++: Semantically enhanced searching and ranking on the desktop". In *Proceedings* of ESWC '06, Budva, Montenegro, pages P.348—362, 2006.

11 Chudoba, K. M., Wynn, E., Lu, M., & Watson-Manheim, M. B. "How virtual are we? Measuring virtuality and understanding its impact on a global organization". Information Systems Journal, 2005, 15, 279-306.

12 Davis, G. B. "Anytime/anyplace computing and the future of knowledge work". Commun '02. ACM 45, 12 (Dec. 2002), 67-73.

13 Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L. and Herlocker, J. “TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers”, In *Proceedings* of the IUI’05, San Diego, CA, USA. ACM Press P.75-82., 2005.

14 Drucker, Peter F. 1999. "Knowledge-Worker Productivity: THE BIGGEST CHALLENGE." California Management Review 41, no. 2: 79-94. Business Source Premier, EBSCOhost (accessed May 9, 2009).

15 Gonçalves, D. and Jorge, J. A. “In search of personal information: narrative-based interfaces”. In *Proceedings* the IUI '08, Canary Islands, Spain. ACM, New York, NY, P.179-188, 2008.

16 González,V.M and Mark, G. “Constant, constant, multitasking craziness: managing multiple working spheres” In *Proceedings* of the CHI '04, Vienna, Austria., ACM, New York, NY, P. 113-120, 2004.

17 Hailpern, J., Jitkoff, N., Warr, A., Karahalios, K., Sesek, R., and Shkrob, N. “YouPivot: improving recall with contextual search”. In *Proceedings* CHI '11, Vancouver, Canada. ACM, New York, NY, USA, 1521-1530.

18 Jensen, C., Lonsdale, H., Wynn, E., Cao, J., Slater, M. and Dietterich, T.G. “The life and times of files and infomation: a study of desktop provenance”. In *Proceedings* of the CHI '10, Atlanta, GA, USA. ACM Press (2010), 767-776.

19 Kidd, A., Adelson, B., Dumais, S., Olson, J. “The marks are on the knowledge worker”. In *Proceedings* of the CHI '94, Boston, MA, USA. ACM, New York, NY, 186-191.

20 "Provenance." Merriam-Webster Online Dictionary. 2009. Merriam-Webster Online. 28 May 2009 http://www.merriamwebster.com/dictionary/provenance

21 Muniswamy-Reddy, K., Holland, D. A., Braun, U., Seltzer, M. “Provenance-aware storage systems”. In *Proceedings* of the 2006 USENIX Annual Technical Conference, June 2006.

22 Rader, E. “The effect of audience design on labeling, organizing, and finding shared files”. In *Proceedings* of the CHI '10, Atlanta, GA, USA. ACM Press (2010), 777-786.

23 Ravasio, P., Schär, S. G., and Krueger, H. “In pursuit of desktop evolution: User problems and practices with modern desktop systems”. ACM Trans. Comput.-Hum. Interact. 11, 2 (Jun. 2004), P. 156-180, 2004.

24 Soules, C. A. and Ganger, G. R. 2005. Connections: using context to enhance file search. SIGOPS Oper. Syst. Rev. 39, 5 (Oct. 2005), 119-132.

25 Tong, H., Faloutsos, C., Gallagher, B., and Eliassi- Rad, T. “Fast best-effort pattern matching in large at- tributed graphs”. In KDD ’07: *Proceedings* of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 737–746, New York, NY, USA, 2007. ACM.

26 Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R." The perfect search engine is not enough: a study of orienteering behavior in directed search". In *Proceedings* of the CHI '04, Vienna, Austria. ACM, New York, NY, 415-422.

27 Tulving, E. and Thomson, D., "Encoding Specificity and Retrieval Processes in Episodic Memory," Psychological Review, Vol. 80, No. 5, 352-373, 1973.

3 The Leyline: A Comparative Approach to Designing a Graphical Provenance-Based Search UI

Soroush Ghorashi, Carlos Jensen

School of EECS
Oregon State University
Corvallis, Oregon, 97331, USA
{ghorashi, cjensen}@eecs.oregonstate.edu

Proceedings of the 2012 The ACM IUI Conference on Intelligent User Interfaces
Lisbon, Portugal
(Under evaluation)

3.1 ABSTRACT

In this paper we explore the design of Leyline, a provenance-based search and file management system, both on a conceptual and user interface level. We perform a comparative analysis and classification of previous provenance based search systems, their underlying assumptions and focus, search coverage and flexibility, and features and limitations. We describe a novel provenance-based search system based on a flexible visual sketchpad interface, and explore how this expands the flexibility of such systems within acceptable limits on complexity and search time. We conclude with design implications and lessons learned in the development and evaluation of such a provenance-based search system.

3.2 INTRODUCTION

The proliferation of cheap and abundant local storage on computers and other devices, as well as the growing trend of cloud storages and computing makes information management and retrieval one of the most challenging tasks facing computer users, and especially knowledge workers. Knowledge workers are defined by the fact that they need access to, process, and produce vast amounts of data, including documents, emails, web pages, etc. With the growth of storage capability, the urge and incentives to store ever more information grows, and traditional incentives to manage, archive and limit storage disappear. Therefore, with a growth in storage capacity comes the potential for a significant increase in the amount of information available, making it difficult to find and retrieve a specific file or resource.

Research has shown that the most effective way for users to find specific files and information on their systems is to carefully arranging them in nested folders and then perform manual searches later (Barreau & Nardi 1995) (Bergman et al. 2008) (Teevan et al. 2004). As the number of resources and nested folders increases, as the number of storage locations increases, and as the number of collaborators trying to impose their own file organization scheme increases, this model breaks down.

One potential solution is to shift the burden onto the computer and use keyword search and indexing tools such as Google Desktop and Microsoft Desktop search. While such tools have the potential to greatly help users find files and information, they are not a universal solution, especially for knowledge workers. Knowledge workers tend to be specialists, working in a narrow sub-discipline and the reuse of information is part of their job description. In this situation, using keywords to find a specific file will often result in a large number of potential hits, and the user will need to manually inspect these to find the right target.

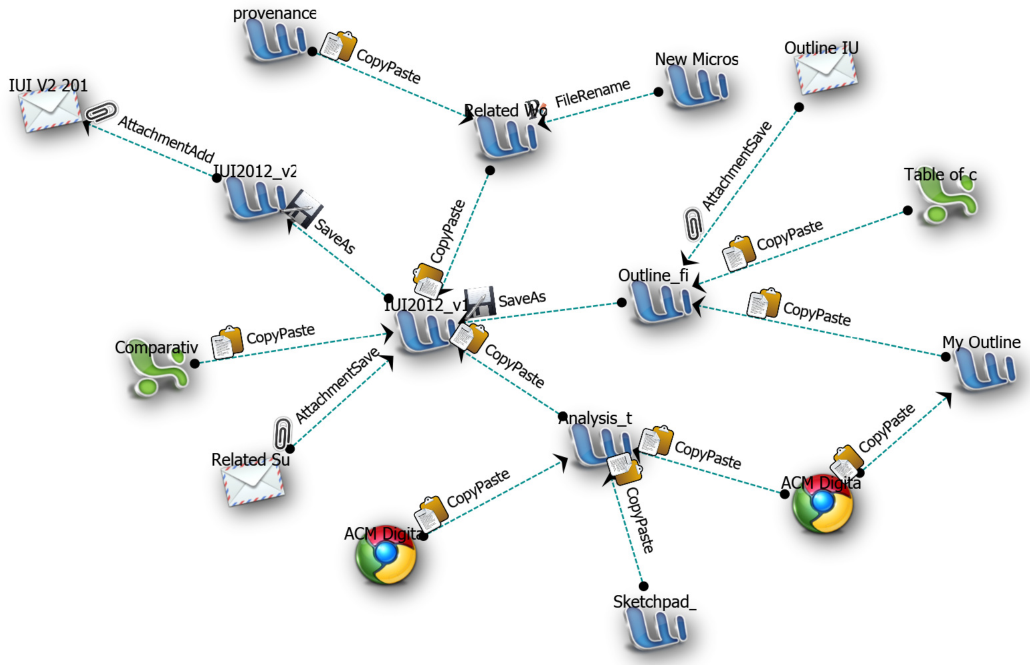


Figure 3.1 Sample provenance graph showing the various resources used in writing this paper.

Because of these limitations, researchers have looked at different ways to improve search. One way of doing so is to use provenance information to enhance keyword search. Provenance differs from other kinds of meta-data because it is based on relation between resources. For example, a user may not remember the exact name of the file they are looking for, or a sufficiently unique keyword, but they may remember that a colleague sent that file in an email. Adding provenance information could not only help identify files more easily, it can also allow us to leverage episodic memory to allow for fundamentally different types of queries.

For instance, provenance data makes it possible to search for inherently non-textual data such as images and audio files by the context of their use or creation. For media such as images, music, video users currently need to remember file names or meta-data such as location or edit and creation dates in order to retrieve them because keyword search systems typically cannot index the content. This does not mean that this is not an active area of research; there are plug-ins for desktop search tools that enable keyword search of music files by accessing repositories of lyrics.

To determine how best to help users include this additional meta-data in search, a number of provenance-based search tools have been developed over the years. These different tools have made use of different types of provenance information, and have been built around a variety of different interface and query models. In this paper we do a comprehensive comparison of these tools to determine their relative strengths and weaknesses, and we introduce a novel new search tool that has:

- A user interface (UI) that allows users to compose query as directed graphs representing provenance events, demonstrated to be effective cues in recall.
- Methods for monitoring provenance events, algorithms to efficiently search using this meta-data, and let users manipulate results to aid recall.

In the next section we will describe relevant related work in this area. We then review previous provenance-based search tools and their advantages and disadvantages, before describing how this analysis leads us to the design of our own system. We discuss our architecture, user interface design choices, and different features. We conclude with a discussion of the challenges to be solved in the design of these types of tools.

3.3 RELATED WORK

Over the last years, a lot of effort has gone into the design and implementation of search tools that use provenance-like information to improve how users search for and find their files and other information on their personal computers. Commercial tools like Google Desktop, Microsoft Desktop Search and Mac OS X Spotlight are primarily based around keyword search, but they also allow user to use some file meta-data to narrow the list of potential results in a textual way. For example in the default Windows search system, user can define the type of resource they are looking for by typing "*type:*" as a keyword in the query, or define a range of dates within which the file was created or used.

One major reason for leveraging provenance in search is that provenance links can be effective memory cues. Research has shown that people easily forget details of computing events, even over short periods of time (Czerwinski & Horvitz 2002), it is hard to remember exactly what we've done after the fact. On the other hand, the psychology literature suggests that contextual information, often associated with provenance events, helps people in later recalling (Tulving & Thomson 1973). In other words the more relevant data users can leverage, the greater the likelihood and ease of recalling something is. In terms of files and documents, provenance relationships have been shown to be effective cues for later retrieval because they help users leverage knowledge of related documents and events. Previous research has also shown that provenance relationships and document reuse is very common among knowledge workers (Jensen et al. 2010), therefore there is abundant data for such systems to potentially work with.

One of the biggest challenges to using provenance in search is capturing such data in the first place. The Provenance-Aware Storage System (PASS) (Muniswamy-Reddy et al. 2006) is a file system specifically designed to monitor and track low-level events that happens to and between files. While no specific tools have been designed to use this information in search, they did introduce a language to query captured provenance information (Holland et al. 2008).

Feldspar (Chau & Myers & Faulring 2008) is a tool that uses associations between files and resources to enhance keyword search. Users can construct their search queries using these associations incrementally and Feldspar updates the results whenever user changes an association in real-time. The user interface is built around a graphical flow-charting metaphor. It relies on meta-data captured by Google Desktop Search.

Quill (Gonçalves & Jorge 2008) is another provenance-based search tool. It has an internal system monitor that tracks and captures meta-data related to documents, email messages, web pages and calendar. Quill uses a more narrative-based query model. Using structured text input, it allows users to describe their target document using pre-define statements and filling in the blanks to predefined narratives.

Contextual cues can also help users in their search process. For example a user might not remember a web page address, but they might remember what music they were listening to while visiting that page. YouPivot (Hailpern et al. 2001) is a tool that allows users to find their desired information by searching for related and memorable activities occurring at the same time.

The concept of provenance as part of a documents' history has been explored further. Lifestreams (Freeman & Gelernter 1996) is built around a timeline metaphor, where all resources and files are tracked and temporally sorted and visualized into a timeline from creation to deletion. By traversing this timeline users can backtrack and see past actions in context, and thus find their resources. Users are able to filter these resources by creating sub-streams, but this interface does not allow for full-fledged search using provenance information.

Stuff I've seen (Dumais et al. 2003) is a search tool that acts very similar to default Windows Search. It first searches its database using keywords, and then ranks the results using contextual cues. Finally it presents results with details such as edit dates, authors, thumbnails etc. Whenever a user changes any filters, results update automatically. Phlat (Cutrell et al. 2006) is another tool that follows same approach. By supporting unified labeling, it allows users to organize their resources and return to them later.

Tools such as Connections (Soules & Ganger 2005) use more traditional keyword input to identify possible resources, and then use temporal locality and context to add other relevant resources to results list. However a newer version of this tool is based on causality relationships because users found these results to be better than locality-based ranking (Soules & Ganger 2006). Beagle++ is another tool that uses the same input method but ranks results based on number of semantic associations between files that are similar to provenance information (Chritia et al. 2006).

DejaView is a personal virtual computer recorder that keeps a record of user activities on the computer (Laadan et al. 2007). It captures application checkpoints and file system state, as well as records contextual information about displayed text. DejaView then allows users to browse and search these records for any visual information that has been displayed on the desktop, and jump to any specific computing state and interact with the desktop as it was at that point. Similarly TaskTracer tracks the context of use of files and applications, though here the goal is to help users recover from interruption by determining which files and resources are associated with what task. This allows users to resume tasks with their context, hopefully restoring their train of thought (Dragunov et al. 2005).

3.4 COMPARATIVE ANALYSIS

In this section we compare different design approaches and tradeoffs made in the design of personal search tools that use provenance-like data either in the formulation of queries, in the search process, or in the presentation of results. For this analysis we included all the search tools we could find, both commercial and academic, that were primarily aimed at helping users search their file repositories. This excluded a number of tools that use provenance information for other purposes. We found a total of five tools: Feldspar, Quill, YouPivot, Stuff I've Seen, and Phlat. We decided to exclude some tools like Google Desktop, which are primarily keyword search tools, augmented to track a limited number of provenance events and expose these in keyword searches. The systems we chose to study here have already been introduced and described briefly in our related work section.

We will now present an analysis of the different design choices made in the design and implementation of these tools. As far as we can tell, no such analysis has been performed before, and there is relatively little knowledge about what design choices have been tried, what has been found to work, and what is the design space yet to be explored. We believe such a categorization to be essential to informing the design of new and better provenance-aware search systems. An overview of the systems studied and their differences can be found in Table 3.1.

Provenance information can be used in a number of different contexts in search tools. Tools can differ in how they gather provenance, what kinds of provenance they track, how they allow users to use provenance in formulating a query, and how they use query in the presentation of potential search results. We finally look at what evaluation has been done of these tools, which is important in order to determine whether these approaches have been adequately evaluated.

3.4.1 Data Gathering

One of the core differentiators between provenance tools is how the provenance information is gathered in the first place. Here systems can be divided into one of three broad families; those integrated into the operating system, more specifically file systems that track provenance events, those monitoring provenance at the application level, and finally the most popular approach, those reconstructing or inferring relations between files from system logs (see Table 3.1 for an overview).

PASS (Muniswamy-Reddy et al. 2006) is an example of first category of tools, those monitoring provenance at the system level. One potential problem with this approach is that it can be hard to either get enough data to make necessary associations, or alternatively, that the data is collected at such a low level that it overwhelms and requires a lot of processing before it can be used in search.

An example of second category, those monitoring provenance at the application level, is TaskTracer (Dragunov et al 2005). TaskTracer captures meta-data about documents, email attachments, web pages, applications and calendar by instrumenting these applications to provide the necessary information. The advantage of this approach is that provenance is captured in a way that is easy to use and understand. The drawback is that it requires instrumenting a large number of applications.

Feldspar (Chau & Myers & Faulring 2008) is an example of the last type of system. Feldspar does not track provenance events, but rather relies on the operating system and other tools, such as Google's Desktop Search to infer provenance events. The advantage of this approach is that it allows users to bootstrap their existing file libraries and add little overhead to everyday operations. The drawback is that this approach makes it difficult to capture popular events such as copy/paste, save as, file renames, etc. This can be a significant problem, as previous research has shown that copy-paste events make up the majority of provenance events (Ghorashi 2012).

Table 3.1 Different provenance-based search tools summary.

Name	Provenance Types	Provenance Monitoring	Provenance Use	UI Approach	Evaluation
Feldspar	File meta-data, keyword, static relations between resources	Extracting relations from Google Desktop's database using its API	Query formulation, Search process	Flow-chart like, List view model (real-time results updating)	Canned data, limited within subjects user study
Quill	Meta-data such as author, storage place, date, physical place tag (home, work, etc.)	Built-in System Monitor to record meta-data about the user's documents, email attachments, WebPages, applications and calendar	Query formulation, Search process	Narrative-based, List of resources' thumbnails (real-time results updating)	Multiple user studies
SIS	File meta-data (such as kind, date, author, email attributes)	Microsoft Desktop Search database, fuzzy matching (car and cars are same), fielded search (author is "john doe")	Query formulation, Search process, Results presentation	Text input with selectable filters, List view of results with a preview and meta-data	Longitudinal study using real data on subjects' PCs (234 people), 6 weeks
Phlat	File meta-data (such as kind, date, author, email attributes). Contextual cues such as user defined tags	Microsoft Desktop Search database, Extra meta-data as tags (Labeling system)	Query formation, Search process, Results presentation	Text input with selectable filters, List view of results with a preview and meta-data	Longitudinal study using real data on subjects' PCs (225 people), 8 months
YouPivot	Environmental factors as contextual cues, user defined marks	Integrated system monitor to record contextual cues and their occurrences	Query formulation, Search Process	Textual input and selectable filters, List view of results	Canned data, limited within subjects user study

3.4.2 Provenance Types

Another way of categorizing provenance-based systems is by looking at what provenance information they use. The more types of relationship they track or use, the more power and flexibility these systems afford the user. As we know from previous research the relative frequency (though not the relative usefulness) of different types of provenance events (Jensen et al. 2010), we can derive a coverage metric (the sum of the relative frequency of all the forms of provenance tracked or used). Table 3.1 gives an overview of the relative coverage of existing systems.

Most of the systems in our sample use relatively simple provenance collection methods, and therefore the forms of provenance they track tend to be relatively basic. We explain these more in depth when we describe the composition of queries and the presentation of results.

3.4.3 Query Composition

Using provenance information in the formulation of queries can allow users to ask fundamentally different questions about their files and activities. When designing query composition interfaces, there is a tension between ease of use and simplicity on one hand, and flexibility and power on the other. The more constrained the interface, the easier it is to use, but the less power and flexibility it gives users.

Except in the case of Feldspar, which allows users to construct queries visually by stringing together conditionals using a flow-charting metaphor, the UI for all the other search tools we found are based on textual input. Feldspar (Chau & Myers & Faulring 2008) allows users to incrementally define associations between different resources by chaining these together. So for instance, you could identify the target as an email, and then select that this email was associated with an event, and that the following people were associated with said event. This allows for very fluid and easily constructed queries.

SIS (Dumais et al. 2003) and Phlat (Cutrell et al. 2006) are based around textual query input. They basically have a query area that users can enter a search phrase into it.

These tools also present a list of filters on meta-data and contextual cues that users can choose and which will be added to query. So, for instance in SIS, you would construct a query by entering some keywords, then select that the target is an email form a document filter, all from the date filter, author etc. In Phlat, the user could annotate the resources, and then use these labels in the search.

Quill (Gonçalves & Jorge 2008) uses a more narrative model, based around a more natural-language way of describing documents. That said, the descriptions must follow strictly defined templates, which are incrementally presented to users, who are basically asked to fill in the blanks. Users have to fill these blanks either by typing or by selecting from a list of alternatives. This approach, though somewhat constraining, represents a more natural and intuitive way to discuss documents, but can be constraining in that users can only perform searches for relationships for which there are templates, or that conform to their own mental models as to how to express relationships.

YouPivot (Hailpern et al. 2001) is a much more limited system than the others, in that it focuses on finding website addresses. It allows users to enter keywords related to their search in a text entry box format, and users can then select from a number of contextual cues, activities they remember doing at the same time. For instance, a user might remember that they were listening to a particular song at the time. YouPivot presents a list of possible context cues based on the keywords the user entered.

3.4.4 Result Presentation

Provenance information can also be used to explain search results, even if it is not used in query composition. Here, the goal is to help users determine which of a number of potential candidate targets is the file they are searching for, or to help jog the users' memory about other things they were doing at the same time or as part of the workflow that produced or used the file in question. Sometimes this can be as useful or more than allowing users to compose more flexible queries.

Feldspar (Chau & Myers & Faulring 2008) presents results in a very basic list view having very standard data (file name, document type, edit dates, etc.) for each item. Any change in the search query makes Feldspar update results dynamically, supporting a trial-and-error approach to search. In this sense, no provenance data is used or presented in the display of results.

Quill (Gonçalves & Jorge 2008) provides little details about search results, relying primarily on the presentation of thumbnails of possible candidate files. While this set of thumbnails gets updated dynamically as the query is modified, the thumbnails themselves can be of little values, for instance if the candidates are all versions of the same document. In these cases it can be very difficult to spot any differences based on thumbnails. Again, no provenance information is used in the presentation of potential results.

SIS (Dumais et al. 2003) presents a snippet of files and emails in the search results in addition to meta-data attributes such as name, date, type, etc. Like SIS, Phlat (Cutrell et al. 2006) has extensive information on results, presented in a list format. In addition to the standard elements, they include any category labels, and allow users to dynamically apply new filters and narrow the results.

YouPivot (Hailpern et al. 2001) also presents results in a list, but it can show these in the context of the users' activities in a stream graph, with image labels related to different activities. In that sense, this is the only tool we've found that makes extensive use of provenance information in the display of results.

3.4.5 Evaluation

In order to determine the soundness or usefulness of an approach, it is important to examine to what extent, and how thoroughly these have been evaluated. Evaluating provenance-based tools can be complicated, in that sometimes we need users to use these tools over an extended period of time before sufficient provenance information has been gathered to be useful in search.

Feldspar (Chau & Myers & Faulring 2008) has been evaluated with a small limited user study of eight participants and using canned data (the file system used was not the users own, but rather one provided by the researchers). The results, both quantitative and qualitative, suggest users' satisfaction in using Feldspar. The interface was intuitive and easy to use, and queries were generally quickly composed, at least with the sample data used. One lingering question is how well the query composition approach scales with the growth of the data set used.

YouPivot (Hailpern et al. 2001) has also undergone a limited user study with seven participants using canned data. Quantitative results showed that users were more satisfied using YouPivot compared to traditional tools, though the same questions remain with regard to the growth of datasets, or the memorability of related events with the passage of time.

Quill's developers have done no evaluation on the effectiveness and performance of their system, but they have done studies measuring the accuracy of stories generated by the tool using users' real documents (Gonçalves & Jorge 2008).

SIS (Dumais et al. 2003) has been evaluated in a longitudinal study with 234 subjects. They used SIS on their personal computers for six weeks and results show that over this time, more users preferred SIS to their traditional search tools. On average, search queries were relatively simple, averaging 1.59 words.

Phlat (Cutrell et al. 2006) has seen a very similar evaluation, with a study lasting for 8 months and involving 225 subjects. Feedback and use data suggests that Phlat is a useful tool and users continued to use this tool during study period. Again, queries were relatively simple, on average 1.60 words.

3.4.6 Summary

As we can see from the data we have found about provenance-enabled search tools, we see that the use of provenance is relatively simple and limited, both in the query composition and results presentation. In the instances where more sophisticated use of provenance has been made, results have been encouraging. In next section we present a novel approach to include provenance information in the composition and presentation of desktop search. We implement this approach in our search system called the Leyline.

3.5 DESIGING THE LEYLINE DESKTOP SEARCH SYSTEM

As we showed in the previous section, most new search tools in this space introduce novel ideas about how to construct queries to incorporate provenance data, but their use has to date been relatively limited. For instance, none of the tools we found used dynamic provenance relationships such as copy/paste, save as, etc. The promise of provenance data is that it makes available more data for users to narrow down possible search results, and that we can leverage episodic memory. This means that we can not only use provenance to potentially enhance the search process, but also use provenance in the presentation of results to help them more easily identify the correct targets, as well as recall other facts about their workflow and the context of their work.

A study at Intel Corporation (Jensen et al. 2010) showed that provenance events are common, memorable and valuable resources for identifying related documents. In most cases, subjects were able to recall more about their files and resources when shown provenance graphs. They found that provenance can connect related resources and that graphs showing such connections are effective cues for recall.

This led us to develop the Leyline along these lines, as a provenance-based desktop search tool using a graphical sketchpad as a primary UI. Users can use provenance data in addition to meta-data such as attributes and keywords to search. Moreover, results are presented in a similar fashion, using a graph representation that makes it easy for user to see all other relevant resources in one place. We describe the design process, and the different features and functionality of our prototype in the following sections.

3.5.1 Architecture Overview

Our system is built on top of the TaskTracer system (Dragunov et al. 2005), which monitors instrumented applications and system events. These events are then exposed to users through a database, which can be queried and combined with more traditional keyword search mechanisms. Figure 3.2 shows an overview of our system architecture.

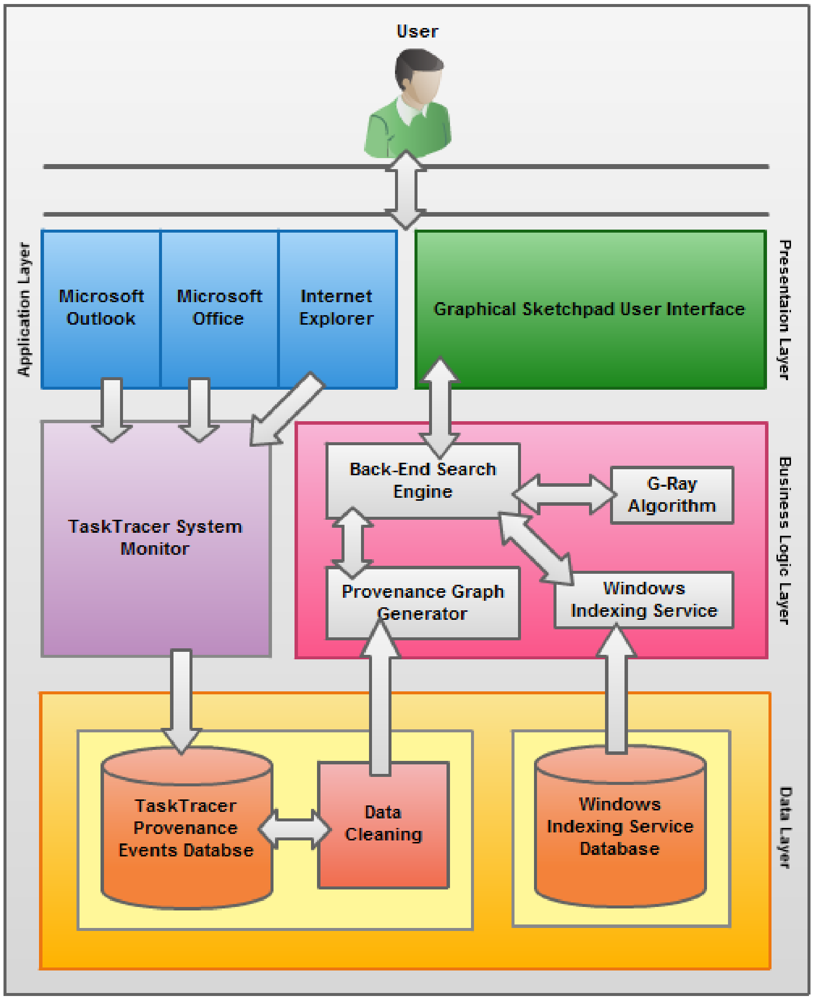


Figure 3.2 Leyline system architecture.

TaskTracer is designed for Microsoft Windows and plugs into the Microsoft Office suite. It records file system operations such as move file, save as, file rename, copy/paste, as well as user activities such as email attachments, file download and file upload. When TaskTracer is running, it records provenance events that triggered by either the user or the system. To make this data usable for search, another component generates provenance graph in an XML format. We'll describe the provenance graph generation later.

There is a UI component that allows users to draw a graph-based query. In this system, resources are nodes, and relations are links between them. The search itself is performed by a different component. This uses a G-Ray algorithm (Tong et al. 2007) to try and match a graph drawn by the user to the set of all provenance graphs described in our XML files. We also integrate Windows Indexing Service to support traditional keyword and meta-data search. Currently our system supports six file types: *Word*, *Excel*, *PowerPoint*, *Email*, *Web* and a catch-all *Unknown* type. Results are displayed in the same way as the queries are drawn, and allow for simple exploration.

Next we describe the different components in more detail, starting with the graphical user interface.

3.5.2 Designing a Graphical Sketchpad User Interface

The longitudinal study at Intel (Jensen et al 2010) showed that provenance information can be an effective cue to trigger user recall. The challenge is how to include these relations in a search query; relationships can be more difficult to describe than keywords. Interviews also showed that directed graphs were effective and intuitive representations of provenance relationships.

For these reasons we decided to use provenance graphs as the cornerstones of the query composition and result presentation UI. This means that in the Leyline, users draw directed graphs in order to compose search queries, and our search engine presents results as graphs that contain not only the target resources, but also the most closely related resources. The challenge with such a UI is designing the system in such a way as to allow both efficient and intuitive query composition. If the UI imposes too much of a burden on users, adoption will suffer.

We followed an iterative design and development approach, with frequent prototype and evaluation sessions. Figure 3.3 shows a sketch of an early paper prototype. In this interface users can draw their search query by adding documents to a canvas. Documents are nodes, and users create relationships between documents by drawing connections between nodes. While intuitive, this early prototype proved too demanding on users' time and effort.

In order to perform preliminary evaluations, we created a representative data-set based on the data reported in (Jensen et al. 2010). We then did an analysis of the average and worst-case query complexity needed to uniquely identify a resource. These findings are presented in (Ghorashi 2012) and proved that the general approach was feasible. We refined our initial model to streamline query authoring and make the system faster and more user friendly. Our final user interface can be seen in Figure 3.6. In the next sections we will discuss individual design elements and decisions.

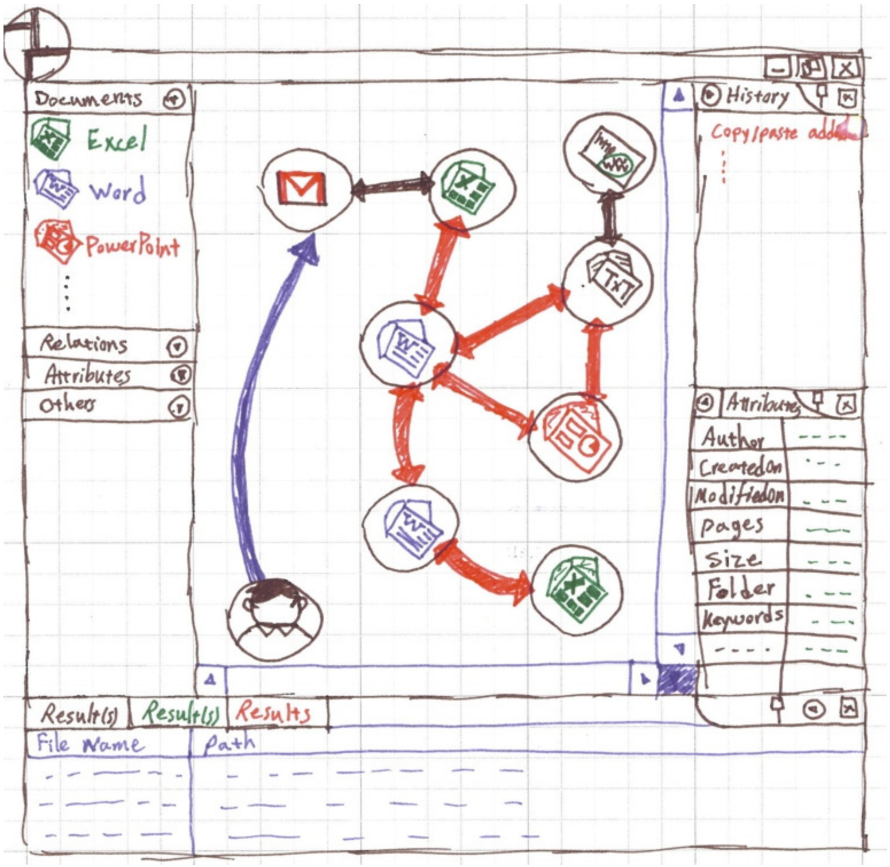


Figure 3.3 Early system concepts.

3.5.3 Query Construction

A user can add a resource to a query by clicking on any blank area of the canvas and set the type of resource using the menu that appears (Figure 3.4.I). If a user wanted to change the type later, they can left click on that resource and the same context menu would appear. Another way to modify resource types is to select the resource and use the properties panel (Figure 3.4.II) in the application window. File types are not required in queries, but help narrow the search space.

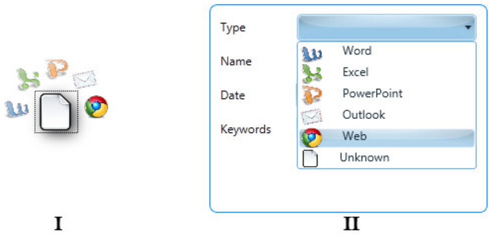


Figure 3.4 Selecting document type using (I) context menu or (II) property pane.

Whenever users want to add a provenance link between two resources, they can click on one (source) and drag the pointer to the other (target) while holding the mouse button down. The type of provenance relation will be set automatically according to the possible types of links given the resources involved and data from the Intel study (Jensen et al. 2010). For example while a link between a web page and a Word document could be a “DownloadFile” relation, it is most probable a copy/paste relation. This simplifies things for users and makes query composition faster and more successful.

Users can change the type of links by right click on the link and using the context menu (Figure 3.5.I) or selecting the link and using properties panel (Figure 3.5.II). Both options only present users with a list of relationships which are possible between the two resources, and the list includes: *Copy/Paste*, *AttachmentAdd*, *SaveAs*, *UploadFile*, *FileRename*, *DownloadFile*, *AttachmentSave*, and a catch-all *Unknown* relation. Relationships are not required in queries, but help narrow the search space.

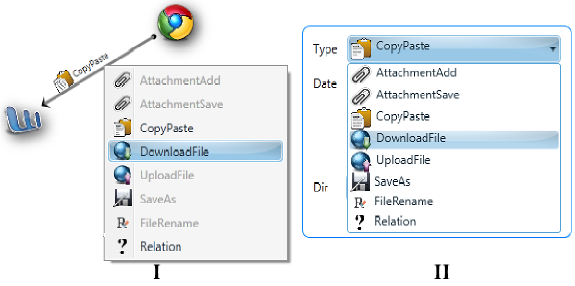


Figure 3.5 Selecting provenance type using (I) context menu or (II) property pane.

In recognition of the general usefulness and power of keyword search, so allow users to associate keywords and partial or complete names with resources using the properties panel. Finally as an optional feature, users can mark which resource they are actually looking for in their query, which allows the system to highlight this in the results presentation.

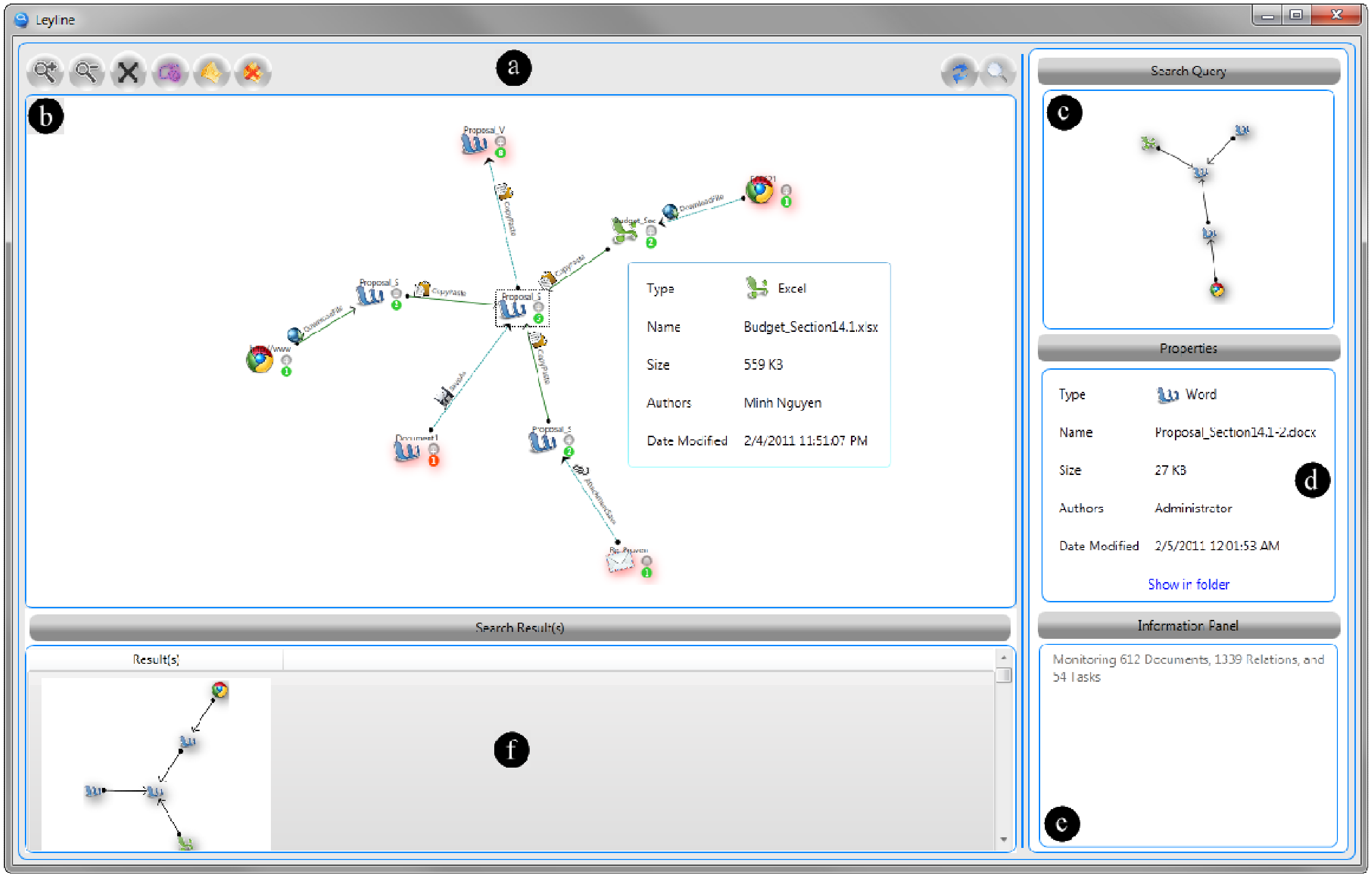
As users construct their query, Leyline monitors their actions and tracks information that will make the search process more exact and efficient. For example if a user adds a date attribute to a provenance link in the query, Leyline filters provenance graph in the database and loads only data related to that date range into the system memory, this will make the actual search faster, more successful and requires less memory.

Having received a graph, the system proceeds to search the provenance database to find relevant matches. This process is similar to solving the np-complete sub graph isomorphism problem. However, introducing “star relations,” a search wildcard allowing a link to potentially be replaced by an arbitrary set of nodes and relationships, we transform the search problem from an exact pattern-matching problem to best-effort matching problem, which reduces the search complexity. This change also has the added benefit of making our system more flexible and forgiving on users and their memory.

3.5.4 Search Process

Having received a graph, the system proceeds to search the provenance database to find relevant matches. This process is similar to solving the np-complete sub graph isomorphism problem. However, introducing “star relations”, a search wildcard allowing a link to potentially be replaced by an arbitrary set of nodes and relationships, we transform the search problem from an exact pattern-matching problem to best-effort matching problem, which reduces the search complexity. This change also has the added benefit of making our system more flexible and forgiving on users and their memory.

Converting exact pattern matching to best-effort matching problem by considering “star relations” allows us to use the G-Ray algorithm (Tong et al. 2007), which does approximate matching in a fast, scalable and efficient way. However, this algorithm works on big undirected graphs where only nodes have categorical attributes. Because edges as well as nodes have attributes in our graphs, we need to pre-process our source and query graphs. We replace each edge with a dummy node with same type and two attribute free links to said node.



Type Excel

Name Budget_Section14.1.xlsx

Size 559 KB

Authors Minh Nguyen

Date Modified 2/4/2011 11:51:07 PM

Type Word

Name Proposal_Section14.1-2.docx

Size 27 KB

Authors Administrator

Date Modified 2/5/2011 12:01:53 AM

Monitoring 612 Documents, 1339 Relations, and 54 Tasks

Figure 3.6 The Leyline interface. (a) Toolbar. (b) Sketchpad. (c) Query window. (d) Properties panel. (e) Information panel. (f) Results panel.

3.5.5 Results Presentation

When graphs are ready we can begin the search process and find best matches. We use G-Ray's goodness function to sort results based on the goodness of the match. If users have provided any meta-data such as full or partial file names, keywords or dates, we do a normal keyword search using Windows Index Service and generate a list of potential matches for each node. Using these lists, we filter the output of the G-Ray algorithm based on whether they contain any of the files found by the Windows Index Service.

Now that results are ready, user see a thumbnail of each candidate result graph in the result panel on the bottom of application window (Figure 3.6.f). Selecting any of these makes the query go to the query panel on the top right of application window (Figure 3.6.c) so the users can more easily compare the results with the query they've constructed. At the same time the selected graph expands in the main canvas and the application goes to result viewing mode where user can only inspect the graph.

If users hover over or select a resource in the result graph, they see more information about that resource (Figure 3.6.b). Doing the same on links, users can see when the provenance event occurred. In addition, we present additional information to assist users in remembering. Having provenance data that tracks the history of all documents, it is not only possible, but likely to feature deleted files in search results. In this case a red dot indicator below and to the right of the document icon is used to indicate that the file is no longer available. Available documents instead have a green indicator (see Figure 3.7). The indicator also works for web pages and emails.

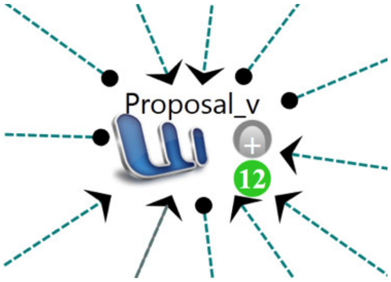


Figure 3.7 A document in result graph that is available on hard disk and has 12 related documents and has been expanded.

We allow users to access resources directly from the graph. Double clicking on an available document or selecting it and clicking "show in folder" from properties panel will open the default file browser with that file pre-selected. Double clicking on a web resource opens the default web browser to that website. Doing the same on available emails finds and opens them.

In order to limit the complexity of the result graphs, we prune them. In order to ensure that users can explore in order to aid recognition or explore workflow, our UI allows users to selectively expand graphs. Each resource in the result graph has a number printed in the availability icon that shows how many provenance links this document has in the full provenance graph. Users can expand the links to each document with a link count greater than zero by right clicking on it and selecting "Expand" or clicking on the plus icon over the availability indicator (see Figure 3.7). This opens up the next level of provenance relationships in the graph. These documents are given a light red shadowed to distinguish them from the documents the search considered the most likely targets of the search. Figure 3.6 shows an example.

3.5.6 Visual Keyword Search Mode

Besides pattern matching search, Leyline also allows users to perform keyword search. While they currently must add a document enter the keyword search mode, they can then add any attribute such as name, type, date, path and etc. to this node to perform a traditional search. In this mode, results are presented in a list view. The main difference between our keyword search and other keyword search tools is that after users find their target file, they can then explore related resources.

We have also integrated Leyline with Microsoft Windows Explorer, which means that users can browse their documents in a traditional way, and whenever they want to find related resources they can right click on the target file and select “Show Provenance” to open Leyline with their file pre-loaded in the main canvas and ready to be explored. This feature can be useful when a user is looking for the source of mistakes, or subsequent uses of their documents.

3.6 DISCUSSION

Leyline is a system monitoring tool capable of recording common and highly memorable dynamic provenance events such as copy/paste, save as, file rename, download file, etc., as well as provide users with an interface for using these to specify file searches. While other similar tools use provenance information, most rely either on static provenance data and file-system events, or infer relationships from system logs. Moreover, Leyline is unique in that it allows users to formulate flexible free-form queries using a mix of traditional keywords and file properties as well as provenance events. There is no constraint at number or type of files or relations in a query.

From our analysis of search tools, we see that Leyline fills a unique niche, and has the potential to fundamentally change the way we approach search and file organization. With the use of provenance information, traditional file organization systems are less important, as context information effectively allows us to find and distinguish between versions and overlapping resources.

The Leyline introduces a number of novel concepts to the desktop search space, the graph-metaphor of representing search queries as well as search results for instance. Another key contribution is the ability to expand and explore provenance graphs to gain deeper understanding of workflow and file history. While we do not know how important this feature is going to be in the single file repository case, we imagine that this feature would be very helpful in the case where groups of people are collaborating and contributing to shared documents. This form of exploration would allow for greater transparency and implicit documentation of the work of collaborators.

That said, important work remains to be done. While we performed running evaluations of our prototypes and based our design on the best available data, we still need to do a longitudinal deployment and evaluation of the system to evaluate its true impact.

We know that there are shortcomings with our system. Our reliance on TaskTracer (Dragunov et al. 2007) to gather provenance information, though convenient for a proof of concept system, means that we are reliant on instrumented versions of any application we wish to track. Switching to a different provenance tracking technology would make it easier to scale and maintain the search application. Preferably such capabilities would be integrated at the operating system level, and the data available to all applications.

In the future we also want to implement dynamic result updates to query modifications, as shown in Feldspar (Chau & Myers & Faulring 2008). We believe this to be a powerful mechanism for encouraging exploration. This combined with our ability to dynamically explore layers of provenance links in results could be a powerful enabler. Having such a feature would also make it easier for users to determine how much detail they need to add to their queries, hopefully allowing them to compose more succinct ones.

3.7 CONCLUSION

It’s hard for users to remember effective keywords to identify unique resource given that much of today’s information work is based around information reuse. A number of different tools have been developed to address this problem based on the inclusion of meta-data about file provenance. We examined the most recent and successful tools in this area. A comparative analysis showed that these tools were relatively limited in their use of provenance information, and how that information is either gathered from users, or how these relationships are represented. We identified a need for more aggressive use of provenance information, as well as the need for a more intuitive representation of said data.

We introduced Leyline, the first search tool that supports dynamic provenance events such as copy/paste, save as, download file, file rename, and etc. in search. This is also the first tool to use a graph-based interface to both express queries as well as present results. Leyline allows users to pose fundamentally different questions about their files, and explore the use and flow of information both n their own practice as well as those of their workgroups. Work needs to be done to evaluate the long-term effectiveness and impact of the Leyline.

3.8 ACKNOWLEDGMENTS

We wish to thank our colleagues in the HCI group at the School of EECS, Oregon State University for their support and help in preparing the paper. We would like to thank our study participants.

3.9 REFERENCES

Barreau, D. and Nardi, B. A. 1995. Finding and reminding: file organization from the desktop. SIGCHI Bull. 27, 3 (Jul. 1995), 39-43

Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N., and Whittaker, S. 2008. Improved search engines and navigation preference in personal information management. ACM Trans. Inf. Syst. 26, 4 (Sep. 2008), 1-24.

Chau, D., Myers, B., and Faulring, A. “What to do when search fails: finding information by association”. In *Proceedings* of CHI '08, Florence, Italy. ACM, New York, NY, P.999-1008, 2008.

Chirita, S. Costache, W. Nejdl, and R. Paiu. “Beagle++: Semantically enhanced searching and ranking on the desktop”. In *Proceedings* of ESWC '06, Budva, Montenegro, pages P.348—362, 2006.

Cutrell, E. Robbins, D. Dumais, S. and Sarin, R. “Fast, flexible filtering with phlat”. In *Proceedings of the SIGCHI ‘06 conference on Human Factors in computing systems*. ACM, New York, NY, USA, 261-270.

Czerwinski, M. and Eric Horvitz, E. An investigation of memory for daily computing events. In HCI 2002, pages 230–245, London, England, 2002.

Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L. and Herlocker, J. “TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers”, In Proceedings of the IUI’05, San Diego, CA, USA. ACM Press P.75-82., 2005.

Dumais, S. Cutrell, E. Cadiz, JJ. Jancke, G. Sarin, R. and Robbins D. C. “Stuff I've seen: a system for personal information retrieval and re-use”. In *Proceedings of the 26th annual international ACM SIGIR ‘03 conference on Research and development in information retrieval*. ACM, New York, NY, USA, P.72-79.

Freeman, E. Gelernter, D. "Lifestreams: A Storage Model for Personal Data," *SIGMOD Rec.* 25, 1 (March 1996), 80-86.

<withheld for review>

Gonçalves, D. and Jorge, J. A. “In search of personal information: narrative-based interfaces”. In *Proceedings* the IUI '08, Canary Islands, Spain. ACM, New York, NY, P.179-188, 2008.

Hailpern, J., Jitkoff, N., Warr, A., Karahalios, K., Sesek, R., and Shkrob, N. “YouPivot: improving recall with contextual search”. In *Proceedings* CHI '11, Vancouver, Canada. ACM, New York, NY, USA, 1521-1530.

Holland, D. A. Braun, U. Maclean, D. Muniswamy-Reddy, K. and Seltzer, M. “Choosing a Data Model and Query Language for Provenance.” In *Proceedings* of the 2nd International Provenance and Annotation Workshop, Salt Lake City, UT, Jun 2008.

Jensen, C., Lonsdale, H., Wynn, E., Cao, J., Slater, M. and Dietterich, T.G. “The life and times of files and information: a study of desktop provenance”. In *Proceedings* of the CHI '10, Atlanta, GA, USA. ACM Press (2010), 767-776.

Laadan, O., Baratto, R. A., Phung, D. B., Potter, S., and Nieh, J. “DejaView: a Personal Virtual Computer Recorder”. *In Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles* (SOSP '07). ACM, New York, NY, USA, 279-292.

Muniswamy-Reddy, K., Holland, D. A., Braun, U., Seltzer, M. “Provenance-aware storage systems”. In *Proceedings* of the 2006 USENIX Annual Technical Conference, June 2006.

Soules, C. A. and Ganger, G. R. 2005. Connections: using context to enhance file search. SIGOPS Oper. Syst. Rev. 39, 5 (Oct. 2005), 119-132.

Soules, C. and Ganger, G. Using context to assist in personal file retrieval. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2006.

Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R.” The perfect search engine is not enough: a study of orienteering behavior in directed search”. In *Proceedings* of the CHI '04, Vienna, Austria. ACM, New York, NY, 415-422.

Tong, H., Faloutsos, C., Gallagher, B., and Eliassi- Rad, T. “Fast best-effort pattern matching in large attributed graphs”. In KDD ’07: *Proceedings* of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 737–746, ACM, New York, NY, USA, 2007.

Tulving, E. and Thomson, D., "Encoding Specificity and Retrieval Processes in Episodic Memory," Psychological Review, Vol. 80, No. 5, 352-373, 1973.

4 CONCLUSION

Users have trouble remembering effective keywords or meta-data to identify unique files or resources on their personal computers when using traditional keyword search tools, given that much work is based on information reuse. Research has been done on augmenting traditional keyword search tools using additional information. In this thesis we investigated the potential benefits of using file provenance data in search.

In the first part of this thesis we showed through the use of real world longitudinal data that provenance events are common and effective to uniquely identify documents in a moderate sized repository. We have also shown that we can build an efficient and intuitive user interface allowing users to build queries which closely matches the way that they reason and think about provenance relationships. This allows users to leverage their knowledge and the recognition rather than the recall both when composing queries as well as evaluating potential results. The search algorithms used in this prototype allows for flexibility, which mimics the kind of flexibility we see in terms of user memory.

In the second part we described and examined different provenance-based search tools. A comparative analysis showed that these tools are relatively limited in their use of provenance data, and how this data is either gathered from users, or how these relationships are represented. We talked about a demand for better use of provenance information, as well as the need for a more intuitive representation of this data. We then introduced Leyline, the first search tool that supports dynamic relationships between different files and resources such as copy/paste, save as, download file, upload file, file rename, and etc. in search. Leyline allows users to ask fundamentally different questions about their files. While a limited user study showed that users liked Leyline's user interface approach both in query composition and results presentation, work needs to be done to evaluate the long-term effectiveness and impact of the Leyline.

5 BIBLIOGRAPHY

Barreau, D. and Nardi, B. A. 1995. Finding and reminding: file organization from the desktop. SIGCHI Bull. 27, 3 (Jul. 1995), 39-43

Bates, M., “Indexing and access for digital libraries and the internet: Human, database, and domain factors", JASIS, 49, pp. 1185-1205, Nov. 1998.

Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N., and Whittaker, S. 2008. Improved search engines and navigation preference in personal information management. ACM Trans. Inf. Syst. 26, 4 (Sep. 2008), 1-24.

Boardman, R. and Sasse, M. A. "Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management”. In *Proceedings* of the CHI '04, Vienna, Austria. ACM, New York, NY, P.583-590, 2004.

Blanc-Brude, T. and Scapin, D. L. “What do people recall about their documents?: implications for desktop search tools”. In *Proceedings* of the IUI, '07, Atlanta, GA. ACM, New York, NY, P.102-111, 2007.

Bureau of Labor Statistics. U.S. Department of Labor. "Occupational Employment and Wages." Press release. Washington, D.C. 1 May 2009.

Chau, D., Myers, B., and Faulring, A. “What to do when search fails: finding information by association”. In *Proceedings* of CHI '08, Florence, Italy. ACM, New York, NY, P.999-1008, 2008.

Chirita, S. Costache, W. Nejdl, and R. Paiu. “Beagle++: Semantically enhanced searching and ranking on the desktop”. In *Proceedings* of ESWC '06, Budva, Montenegro, pages P.348—362, 2006.

Chudoba, K. M., Wynn, E., Lu, M., & Watson-Manheim, M. B. “How virtual are we? Measuring virtuality and understanding its impact on a global organization”. Information Systems Journal, 2005, 15, 279-306.

Cutrell, E. Robbins, D. Dumais, S. and Sarin, R. “Fast, flexible filtering with phlat”. In *Proceedings of the SIGCHI ‘06 conference on Human Factors in computing systems*. ACM, New York, NY, USA, 261-270.

Czerwinski, M. and Eric Horvitz, E. An investigation of memory for daily computing events. In HCI 2002, pages 230–245, London, England, 2002.

Davis, G. B. “Anytime/anyplace computing and the future of knowledge work”. Commun '02. ACM 45, 12 (Dec. 2002), 67-73.

Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L. and Herlocker, J. “TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers”, In *Proceedings* of the IUI’05, San Diego, CA, USA. ACM Press P.75-82., 2005.

Drucker, Peter F. 1999. "Knowledge-Worker Productivity: THE BIGGEST CHALLENGE." California Management Review 41, no. 2: 79-94. Business Source Premier, EBSCOhost (accessed May 9, 2009).

Dumais, S. Cutrell, E. Cadiz, JJ. Jancke, G. Sarin, R. and Robbins D. C. “Stuff I've seen: a system for personal information retrieval and re-use”. In *Proceedings of the 26th annual international ACM SIGIR ‘03 conference on Research and development in information retrieval* . ACM, New York, NY, USA, P.72-79.

Ghorashi, S. Jensen, C. “Provenance-based search using graphical sketchpad”. In *Proceedings* of the CHI '12, Austin, TX, USA.

González,V.M and Mark, G. “Constant, constant, multitasking craziness: managing multiple working spheres” In *Proceedings* of the CHI '04, Vienna, Austria., ACM, New York, NY, P. 113-120, 2004.

Hailpern, J., Jitkoff, N., Warr, A., Karahalios, K., Sesek, R., and Shkrob, N. “YouPivot: improving recall with contextual search”. In *Proceedings* CHI '11, Vancouver, Canada. ACM, New York, NY, USA, 1521-1530.

Holland, D. A. Braun, U. Maclean, D. Muniswamy-Reddy, K. and Seltzer, M. “Choosing a Data Model and Query Language for Provenance.” In *Proceedings* of the 2nd International Provenance and Annotation Workshop, Salt Lake City, UT, Jun 2008.

Jensen, C., Lonsdale, H., Wynn, E., Cao, J., Slater, M. and Dietterich, T.G. “The life and times of files and information: a study of desktop provenance”. In *Proceedings* of the CHI '10, Atlanta, GA, USA. ACM Press (2010), 767-776.

Kidd, A., Adelson, B., Dumais, S., Olson, J. “The marks are on the knowledge worker”. In *Proceedings* of the CHI '94, Boston, MA, USA. ACM, New York, NY, 186-191.

Laadan, O., Baratto, R. A., Phung, D. B., Potter, S., and Nieh, J. “DejaView: a Personal Virtual Computer Recorder”. *In Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles* (SOSP '07). ACM, New York, NY, USA, 279-292.

Muniswamy-Reddy, K., Holland, D. A., Braun, U., Seltzer, M. “Provenance-aware storage systems”. In *Proceedings* of the 2006 USENIX Annual Technical Conference, June 2006.

"Provenance." Merriam-Webster Online Dictionary. 2009. Merriam-Webster Online. 28 May 2009 http://www.merriamwebster.com/dictionary/provenance

Rader, E. “The effect of audience design on labeling, organizing, and finding shared files”. In *Proceedings* of the CHI '10, Atlanta, GA, USA. ACM Press (2010), 777-786.

Ravasio, P., Schär, S. G., and Krueger, H. “In pursuit of desktop evolution: User problems and practices with modern desktop systems”. ACM Trans. Comput.-Hum. Interact. 11, 2 (Jun. 2004), P. 156-180, 2004.

Soules, C. A. and Ganger, G. R. 2005. Connections: using context to enhance file search. SIGOPS Oper. Syst. Rev. 39, 5 (Oct. 2005), 119-132.

Soules, C. and Ganger, G. Using context to assist in personal file retrieval. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2006.

Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R.” The perfect search engine is not enough: a study of orienteering behavior in directed search”. In *Proceedings* of the CHI '04, Vienna, Austria. ACM, New York, NY, 415-422.

Tong, H., Faloutsos, C., Gallagher, B., and Eliassi- Rad, T. “Fast best-effort pattern matching in large attributed graphs”. In KDD ’07: *Proceedings* of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 737–746, ACM, New York, NY, USA, 2007.

Tulving, E. and Thomson, D., "Encoding Specificity and Retrieval Processes in Episodic Memory," Psychological Review, Vol. 80, No. 5, 352-373, 1973.