VisArchive: A Time and Relevance Based Visual Interface for Searching, Browsing, and Exploring Project Archives (with Timeline and Relevance Visualization)

by

Keyun Hu
B.Sc., University of Victoria, 2006

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Supervisory Committee

VisArchive: A Time and Relevance Based Visual Interface for Searching, Browsing, and Exploring Project Archives (with Timeline and Relevance Visualization)

by

Keyun Hu
B.Sc., University of Victoria, 2006

Supervisory Committee

Dr. Melanie Tory, (Department of Computer Science)
Supervisor

Dr. Sheryl Staub-French, (Department of Computer Science)
Departmental Member

Abstract

Supervisory Committee
Dr. Melanie Tory, (Department of Computer Science)
Supervisor
Dr. Sheryl Staub-French, (Department of Computer Science)
Departmental Member

Project file archives are becoming increasingly large. The number of files, information and data that need to be created, accessed and modified throughout a project can be overwhelming. It is critical for project participants or contributors to find relevant information in project archives quickly. In this thesis, I present VisArchive, an interactive visualization tool that provides users with better awareness of search results within project archives. VisArchive visualizes the relevance-ranked search results with a color-coded stacked bar chart and interactive timelines and provides supporting visual cues to help differentiate search results based on searched keywords. It aims to allow users to interactively search, browse, and explore information in project archives, including access history, effectively and efficiently. I will present two case studies to illustrate how VisArchive can be used to support searching, browsing, and exploring information in building construction and open source software projects. In addition, I discuss how VisArchive can be improved to address information retrieval problems and work across different domains. VisArchive demonstrates the combination and application of several visualization techniques to the problem of searching and navigating project archives.

Table of Contents

Supervisory Committee ii
Abstract iii
Table of Contents iv
List of Figures v
Acknowledgments vi
Chapter 1: Introduction 1
1.1 Thesis Outline 5
Chapter 2: Related Work 7
2.1 Timeline-based interface for showing search results 8
2.2 Visual indication of search relevance and which search keywords are matched 11
2.3 Visual representation of file access logs 13
2.4 Summary 15
Chapter 3: Visualization Design and Rationale 16
3.1 Design Objectives 16
3.2 Prototype Design 21
3.2.1 Overview of the Prototype 21
3.2.2 Interactive Timelines and Visualization of the Search Results 23
3.2.3 Information Browser and Visual Cues Supporting the Search Results 27
3.2.4 Advanced Filters 30
3.2.5 Access History Visualization Viewer 32
3.3 Algorithm for Generating and Visualizing Search Results 34
3.4 Implementation 36
Chapter 4: Case Studies 38
4.1 Case Study: Construction Project File Archives 39
4.1.1 Searching Files that Match all the Search Keywords 40
4.1.2 Exploring Files relevant to the Search Keywords 43
4.1.3 Exploring File Access History 46
4.2 Case Study: Defects Tracking of Mozilla Thunderbird Project 48
4.2.1 Searching Defects in the Software Defect Archives 50
4.2.2 Exploring the Defect Archive and Relevant Software Defects 52
Chapter 5: Discussion 56
5.1 Timeline based interface for showing search results 56
5.2 Visual indication of search relevance and which search keywords are matched 58
5.3 Visual representation of file access 59
5.4 General Discussion 60
5.5 Improving on the VisArchive prototype 62
Chapter 6: Conclusion and Future Work 64
Bibliography 65

List of Figures

Figure 1. Overview of VisArchive's main interface.. 4
Figure 2. Timeline visualization while searching with two keywords. 25
Figure 3. Timeline visualization while searching with four keywords............................ 25
Figure 4. Color scale representing the levels of relevance 26
Figure 5. Information browser including description viewer displaying file items for a four keyword search... 28
Figure 6. Advanced filters designed for the construction file archives 31
Figure 7. Visualization showing access history of a selected file.................................... 33
Figure 8. Diagram of generating relevance-ranked oriented search results...................... 35
Figure 9. Timeline visualization while searching keywords: electrical, mechanical, structural, and Mike 42
Figure 10. Information browser displaying the files with color-coded visual supports ... 43
Figure 11. Visualization of access history filtered by selected access users 47
Figure 12. Conventional list of search results provided by Bugzilla................................ 50
Figure 13. Visualization of search results provided for searching "message compose window" in the software defect archive.. 52
Figure 14. Visualization of search results provided for searching "compose window file attachment" in the software defect archive 53
Figure 15. Sample information displayed in information browser 54

Acknowledgments

First, I would like to express my heartfelt thanks to my supervisor, Melanie Tory, for her intelligence and guidance in my career as a graduate student, especially for her understanding, support and encouragement during the most difficult time in my life when I lost my mom in 2009.

Thanks to my colleagues in the VisID (Visual Interaction Design) lab for their help, creativity, especially their inputs on inspiring my studies.

Thanks to the members of my committee, Dr. Sheryl Staub-French, for her expertise and guidance on improving my thesis. Thanks to Dr. Stephen W. Neville, for his time being my examiner and recognizing my work.

Finally, I am forever grateful to my parents, especially my mom even when she became ill, for their love, generosity, support and understanding throughout my graduate school. May my mom rest in peace and be happy for my accomplishment.

Chapter 1: Introduction

Nowadays in project management, electronic data storage and database management systems offer simple and inexpensive ways to store electronic documents generated through a project, and they also enable people or software tools the ease and capacity to access the information remotely from anywhere. For example, in a construction project, documents such as meeting agendas, meeting minutes, schematic diagrams and computer-aided design (CAD) drawings contain rich information critical to project success. This information and documentation is often generated digitally (but could also be scanned) so that it is easy to archive it in a shared digital storage repository. Similarly, in a large software project, thousands of software defects can be reported and generated over time by software testing specialists. These defect items are typically stored in a defect management system so that they can be accessed by software developers who are in charge of fixing the defects. In this thesis, I define a *project archive* as a collection of files or information being generated or recorded historically through the project and stored in a common shared repository.

My research was motivated by a common problem encountered in the construction industry. Like other domains, construction project success depends on the capacity of individuals to rapidly retrieve and manipulate information from an archive containing vast and highly diverse documents [37], such as schematic drawings, cost data, schedules, meeting records and code requirements. Although this scattered information can be chronicled and archived in a common repository accessible to all stakeholders or even integrated into a database management system for higher-level data processing, the increasing amount of information and the increasing complexity of its structure make searching and exploring information in the project archive challenging and time-consuming. My research was motivated by my research group's involvement in a building project called the Centre for Interactive Research on Sustainability (CIRS) (details described in Chapter 3). The study conducted by Melanie Tory et al. for the CIRS project [37], found that individuals had a difficult time searching and locating files in the Buzzsaw archive unless they were already familiar with the hierarchy structure and the name of the item they were searching for. In the CIRS building project, a third-party application called Buzzsaw [42] was used as a central repository by the building design team for information archiving, sharing and retrieval. The project documents archived in Buzzsaw were organized and stored in hierarchical directories. This allowed individuals to access files by browsing directories, and search files by meta-data (e.g. keywords, date, authors). However, Buzzsaw did not provide a mechanism to allow individuals to amass the knowledge of the file structure, which was necessary to quickly navigate and locate the relevant documents. It also did not allow users to understand how relevant the listed search results matched their searched keywords, such as to determine the number of searched keywords and/or to identify the searched keywords matched in the relevant documents. For example, when searching documents with keywords “mechanical, electrical, structural”, an architect would not be able to distinguish or group documents that contained “mechanical electrical” and “electrical structural”. Instead, individuals needed to read the textual information (e.g. textual meta-data, document content) of each file in order to know the keywords contained within it. This could become a tedious and time-consuming process when using Buzzsaw.

In addition, better understanding of file access history can be useful for project individuals updating recently used files, preparing project meetings, marking or deleting out-of-date files, etc. Buzzsaw enables individuals to track and manage versions by viewing activity logs (access history) of a document. However, the access history is conventionally displayed as a list of activities, in which the temporal information of the activities and quantitative information of each access type (e.g. created, accessed, modified, etc.) are difficult to perceive. A visual representation of this information will help individuals better understand the file access history more efficiently. I will elaborate on this design motivation in Chapter Three.

In order to enhance the management and accessibility of the information in the project archive, my primary research goal was to develop a new interface approach that would enable people to search, explore and access relevant information from the archive effectively and efficiently. Since visualization techniques can present objects and their relationships visually, thereby offloading cognitive effort onto the perceptual system [1], my research focuses mainly on using visualization techniques to support searching, browsing, and exploring the extensive and complex collections of data in a historical project archive. In this thesis, I introduce VisArchive, an interactive tool that visualizes a project archive and search results by time and relevancy of the search keywords. It aims to provide users the capabilities to search, browse, and explore the information, including access history of a particular file. VisArchive employs a combination of timeline visualization, color-coded stacked bar chart and additional supporting visual cues, which I anticipate will be easily understood by visualization novices. In general, VisArchive offers three key design ideas: (1) Organizing the project archive and search results using a timeline-based layout, (2) Visually representing search relevance and which search keywords were matched, and (3) Visually representing the file access history. Figure 1 shows the main interface of VisArchive that presents the search results when a user searches using a combination of keywords. VisArchive visualizes the relevance of the search results with color scale on bar charts. The bars are shown on multi-scale timelines (Figure 1(d)) that enable users to find and explore relevant results more easily. VisArchive also allows users to identify which search keywords were matched by mapping colors to the search keywords in the file browser (Figure 1(b)). The design and implementation of the interface will be described in Chapter 3.

Figure 1. Overview of VisArchive's main interface

(a) Search bar; (b) Information browser; (c) Description viewer; (d) Interactive timelines; (e) Time slider; (f) Time range; (g) Time range selector.

The design ideas from my prototype are not only applicable to construction projects but also to search tasks in project archives of other domains. In my studies, rather than comparing the performance of my prototype to existing tools, I conducted two case studies to evaluate the feasibility of my design ideas for two different domains. I focused primarily on understanding what value the three design ideas could provide for supporting search tasks within each domain.

In this thesis, I demonstrate the design details of my prototype, including the relevancy-based algorithm, and the case studies that I conducted to evaluate the prototype in two different domains. Moreover, I discuss how the prototype could be enhanced to support other applications. This research demonstrates the combination of several visualization techniques to the particular application of searching and navigating project archives.

1.1 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter 2 - Related Work: Reviews research literature related to my prototype design.

Chapter 3 - Visualization Design and Rationale: Describes the details of design goals, design ideas, algorithm and implementation of the prototype.

Chapter 4 - Case Studies: Demonstrates the feasibility of my prototype with two case studies of a construction project archive and a software defect archive respectively.

Chapter 5 - Discussion: Discusses the core design ideas, evaluation, generalization and potential improvement of the prototype.

Chapter 6 - Conclusion and Future Work: draws conclusions from the research and discusses possible future research directions.

Chapter 2: Related Work

The *Berrypicking* model of information retrieval has been applied to the design of early search interfaces [3] to improve information seeking and navigation. The model indicates that information searchers constantly change search terms and search direction based on the results returned and they need to continuously explore new results. Information foraging theory [39] proposed by Pirolli and Card describes information retrieval behaviour relevant to the design of tools for information seeking. They evaluated people's visual information foraging in a *focus + context* visualization [40] providing a better understanding of how information visualization can be affected by factors such as information scent and visual density. Since Ahlberg *et al.* developed the concept of Dynamic queries [11] and the principles of Visual Information Seeking [8], many researchers and developers have developed graphical widgets and visualization techniques to support browsing, searching and visual scanning to identify results. For example, Sliva *et al.* [16] described a visualization tool that implemented the principles of Visual Information Seeking to assist the exploration of large collections in digital libraries.

I took the *information foraging theory* and *berrypicking* model into account in designing my prototype with high-level navigation and visualizations to enable users to more easily search and access relevant information in project archives. My prototype aims to provide effective visual *scent* of the relevant information through the visual indication of search relevance and matched keywords. The multi-scale timeline-based interface aims to enable users to browse and access the *focus + context* visualization of *scent* to the relevant information with high visual density. In addition, my prototype implements and extends *Dynamic Queries* and *Visual Information Seeking principles* including updating and visualizing search results based on the search queries, and integrating filters, visual displays and additional utilities to support searching, browsing and exploring the information space. As a result, VisArchive should enable users to interactively search, filter, browse the archive and identify relevant information more effectively and efficiently.

Based on the three design ideas from my VisArchive prototype, my related work contains the following topics: timeline-based interfaces for showing search results; visual indications of search relevance and which search keywords are matched; visual representations of file access. I discuss each of these in turn.

2.1 Timeline-based interface for showing search results

Timelines have been widely used in variety of applications to visualize and present historical and temporal data when time is a supportive part of information retrieval. Interactive timelines[7] can improve user interaction with data by allowing a viewer to scroll, change scale, select from multiple timelines, and display attributes of events. tmViewer [6] enabled users to explore temporal metadata, relationships in digital libraries or databases with an interactive timeline, but it focused on displaying metadata on the timeline in addition to temporal relationships. Indratmo *et al.* [4] developed a new visualization tool called iBlogVis that provided rich overview and enabled users to visually browse a blog archive by using a time slider. Studies have also shown that a timeline can be used as an interactive filter for information indexed by time [10, 30]. Other examples of timelines, including Lifelines [5], KNAVE-II [25], Themail [27], CodeSaw [28], PatternFinder [22] and Archive-It [9], provided interactive environments to visualize data sets along timelines across different domains. Although these previous studies show that timelines can be an effective way to visualize time-oriented data sets, they are limited to focusing either on presenting the statistical information retrieved from the data sets, supporting information browsing or serving as a time-oriented information filter. In contrast, VisArchive focuses on applying a combination of visualizations such as multi-scale and multi-dimensional interactive timelines, color-coded visual supports, color scale visualizations, and filters to support searching and exploring historical project archives. Moreover, the timeline interface is the main interface that a user will interact with and perform search and exploration tasks.

The Perspective Wall [17] integrated detail and context views of information through a timeline visualization by presenting the detail view panel in the centre and folding two perspective panels with related context on either side. This method has a limitation for visualizing large scale datasets. For example, perceiving information context from the folded visualization panel becomes difficult when the amount of data increases and more visual cues are applied. The multi-scale timeline slider [18] effectively visualizes the detail information of the selected period while also retaining the entire context. However, archive datasets with textual information such as file names are difficult to represent on the multi-scale timeline slider alone unless the number of items focused on is within a small limit and textual information is avoided.

Continuum [24] is a multi-scale timeline visualization tool that visualizes large-scale datasets with temporal data. It represents faceted temporal data with the capability to explore its hierarchical relationships on the timeline. It also enables control over the detail of level to be represented. The visualization focuses on temporal events which have start and end points. However, it does not support keyword-based interactive search tasks, which focus instead on enabling visual exploration of relevant search results and navigation to relevant pieces of information. VisGets [19] enables searching, filtering and exploring online news items in different ways including through timeline and keywords. The statistical and relevance information of the search results are visualized on timelines to support exploring the news items. However, the timelines can only be used as an interactive filter; the user is not allowed to scroll through the timeline for quickly browsing the results in different time periods. In addition, the news items in the results browser are not organized in a timeline and temporal information is missing when a user browses the results. The user has to open each single item to acquire the temporal information.

To effectively visualize historical archives and search results and support users' browsing and navigation tasks, my timeline-based interface integrated three interactive timelines as *focus + context* visualizations: two horizontal multi-scale timelines that provide an overview and a detail view (with selected time range) of visualization of relevant search results over the archives, to support users in identifying relevant results; and a vertical timeline-based information browser that provides textual information of archive elements and visual indications of relevance and matched keywords. The information and visualizations shown in all the timeline containers are updated synchronously, and the user can interact with the timelines by using a time slider as a time range filter on the horizontal timelines. VisArchive contributes to building a novel interactive timeline-based search interface by reusing existing best practices of timeline visualizations and interaction techniques to solve specific project archive information retrieval problems. In addition, visualization of keyword-based search relevance on the timeline interface is also a major contribution of VisArchive.

2.2 Visual indication of search relevance and which search keywords are matched

Previous studies have evaluated different visualization interfaces for information seeking tasks. NIRVE [14] supports visualization of search results in Text, 2D, and 3D interfaces and a study showed that users' performance was affected by many factors such as computer skills and type of search task. The tool visualizes search results and their relationships on 2D or 3D map views with lines linking the related items and allows users to explore the information of each item by selecting and expanding the item. However, it will be difficult to interact with the items and view information within the items when the number of items grows very large. Foo and Hendry [13] created and evaluated a suite of visualizations for searching one's desktop by keywords. Relevant results to the search keywords and filters were categorized by using different colors, shapes, etc. so that users could effectively identify and distinguish the results relevant to different searched keywords and filters. However, Foo and Hendry did not focus on searching temporal data and information. Moreover, the visualizations were presented in a single view which is not effective to support focus + context type of search tasks. For example, when a user zoomed in to explore the detail view of one area of the visualization, the user lost the context from the single view display.

Visualization of the search relevance and ranking helps users prioritize the relevant information. Veerasamy *et al.* [15] designed a search tool that shows the number of results matched, and the rank, to each sub-query by providing a visualization of sub-queries that were relevant to the searched keywords. However, this tool did not effectively support searching temporal information, especially when it is important for information retrieval such as in a project archive. Similan [21] used color-coded visual cues to match each event to effectively support visual comparison of students' records. My prototype integrates similar visualizations in the item browser for identifying matched keywords of each item; however, I aim to more efficiently support users in identifying which search keywords each item matched.

"Brushing and linking" [41] has become a standard technique for relating multiple views in information visualization. Color-coded visualizations [12, 20, 26, 30] have been used for different purposes to classify or group information that are similar so that it can be visually recognized by users. Perhaps most relevant is, Cambiera [38], which allowed users to visually connect different groups of data or activities and updates in different visualizations for a visual analytics task. Specifically, Cambiera colour-coded search keywords to indicate what keywords had been used and which ones matched each document. However, Cambiera focused on providing awareness of users' analytical activities to others in a collaborative search task, which is different from my objective. VisGet [19] used color-coded weighted brushing to indicate search results with different relevance mapped to the keywords. However, the visualization did not visually distinguish results with the same relevance ranking but different matched keywords. Jones et al. [30] suggested a process for creating data visualizations in collaborative engineering projects by constructing a text visualization task taxonomy and creating visual mappings of the text data. Example visualizations include using colors to encode the importance and textual classification of emails and bar charts to illustrate the frequency of emails, useful for prioritizing tasks and activities. However, the visualization design process does not include interactive searching, browsing and retrieval of information.

By providing a visual representation of relevance, users should be able to identify and discover relevant documents more easily and effectively. My prototype provides visual representations and visual cues to interactively visualize the relevance-ranked search results and assist users in visually identifying which keywords were matched by each document. In addition, I use an extra visual cue to indicate the locations of search results that matched all the keywords. This enables user to identify the most relevant items rapidly. I also designed the visualization of relevance-ranked search results to be displayed over the timeline interfaces along with the entire project archive so that it can bring better insight and better understanding of the context information (e.g., relevance related to other files, file importance in overall project timeline) to the users. This will be elaborated and discussed in the following chapters.

2.3 Visual representation of file access logs

Similar to the type of access history information, previous work has shown that visualization can be used to explore temporal user activities, events, personal records, etc. Lifelines [5, 23] visualizes medical records of a patient such as past symptoms, diagnoses and medications through an interactive timeline-based interface, and aims to enhance navigation and analysis of the records. Patient's records contain many types of activity which can be similar to file access activities. However, Lifelines did not classify the type of activities. My prototype focuses on the file access activities that can be grouped into a limited number of types that can be visualized differently (e.g., file access history actions can be classified as file created, accessed, modified, etc).

The Timespace [2] visualization system provided overviews of user activity on multiple projects and detailed views of user activity within a selected project, allowing users to explore the activities on the projects. However, the tool focused on personal activities and did not support exploration of group activities (e.g., who has modified a file on a specific date). Augur [29] visualized software artifacts and development activities with color-coded indications over the source code, allowing developers to explore the relationships between the artifacts and activities. However, the target population of the visualization was limited to software developers with programming knowledge. PragmatiX [31] provided a visualization of collaborative change logs, to help managers monitor progress, tracking and exploring quality-related issues such as overrides and coordination among contributors. It focused on change log analysis and exploration. By contrast, my access log visualization interface focuses on supporting information retrieval tasks.

Little previous work focuses specifically on visualizing file access logs. However, this feature can assist users in searching and exploring temporal shared project archives, in which the file or information may be accessed by many others throughout the project timeline. For example, a project manager might want to confirm that an identified file is really what he needs by asking additional questions such as whether the file has been reviewed and modified by the group of architects and when. My prototype visualizes the access history in a combination of timeline-based interfaces, color-coded visual mapping and filters which are similar to my main interface for searching relevant items in the archive. It allows users to interactively browse the history timeline and filter the access information by other users' names.

2.4 Summary

In this chapter, I presented previous work with regard to the design of VisArchive. Since VisArchive is designed for users who are mostly data visualization novices, I use a combination of easy-to-understand visualizations and interaction techniques to solve the information retrieval problems in a novel and possibly more effective way by combining the three design ideas. Different from previous work, VisArchive provides a novel experience in searching, browsing and exploring information in project archives including visualizing search results with relevance over multi-scale timelines, matched search keywords, and access history for project archive exploration. The design and implementation of the prototype are described in the following chapter.

Chapter 3: Visualization Design and Rationale

In the early study conducted from the CIRS building project, one of the common bottlenecks identified were the time demands of searching for documents and relevant information and the inefficiency of the system used [37]. For this reason, my design goal was to enable users to search, browse and explore artifact archives more easily and effectively. Specifically, I aimed to allow users to browse and explore the electronic documents, items or information relevant to the search results in the archives. My project was motivated by a common problem encountered in the construction domain, in which electronic documents for a project are archived and stored over time in a central repository. I also considered other domains that have non-spatial, metadata-based and time-oriented data, such as source files of a software project, or electronic entries of medical records in a database. My target users include, but are not limited to, project managers, project engineers, software developers, and doctors — who need to search and access information from archives across different domains. In this chapter, I will describe my prototype design in detail.

3.1 Design Objectives

My prototype was motivated by what Melanie Tory et al. had identified through an ethnographic field study [37] as a common problem encountered in construction projects: namely, that information seeking and retrieval from a large shared information archive could be difficult and time-consuming. The study found that project members had difficulty searching and locating relevant documents when they did not know where to find the information. For example, when a mechanical consultant was asked to locate the images of water filtration systems on his laptop during one observed project meeting, the consultant spent ten minutes searching for the images. The digital files of the construction project were archived into different directories organized in a hierarchy and stored in a central shared repository. These files could be accessed by browsing the hierarchical directories within an existing software application for project management, specifically Buzzsaw. However, project members had some flexibility in saving the digital files in different directories and they often created their own designated space for file storage and sharing. Consequently, it was difficult for individuals to search for information that they were not familiar with and challenging to explore the project archive with the existing tools. The construction experts were interested in an effective and efficient way to find information and more importantly to explore relevant information in the project archive.

In a construction project, time spent on searching document files, especially finding the files containing the information related to a specific topic, can be costly and sometimes may disrupt the individual's or group's work process. In the example described above, the ten-minute disruption to the meeting in order to search for the file negatively affected meeting productivity, created bottlenecks and interrupted the discussion, all of which can be costly to the project. Imagine a different scenario in which one project manager is making a construction claim for his company because of a labour dispute that occurred during the construction period, and which had led to multiple delays in completing the structure — delays that had not been allowed for in the construction contract. To prepare for the claim, he needs to find all the files for construction of the unit in question, including design documents, meeting minutes, schedules and contracts. Not only is the search time-consuming, but also wrought with impediments that prevent him from finding everything he needs. Some of the documents have been modified since he saw them last; he cannot find minutes for the meetings he had missed; he does not need to see land registry or tax files for the project, yet was unable to filter these out of the search results; his colleagues have filed certain items in different categories than he would normally put them, and there is no tracking mechanism to see who has modified which file. Finding the required documents and identifying who has accessed or modified them can be a tedious and time-consuming process for any project manager, especially when he or she is not only person maintaining those documents. More importantly, making a claim with missing relevant information is insufficient, and unlikely to reach a favourable outcome for the innocent party. With all the files and documents related to a project archived and centralized in a shared repository, ensuring individuals can rapidly retrieve and access relevant information — when they need it — is vital to the success of the repository.

To achieve better file search efficiency, information filtering is an important feature for the user while they are searching the archive — filtering out the unnecessary information will provide more accurate search results. For example, a user may only want to find PDF documents from the archive; a manager may want to find the documents created by particular persons. In the case of the project manager searching for files to support his construction claim, information filtering would have enabled him to filter out the tax and land files that he did not need.

In addition, viewing access history can be useful for individuals. Since the files and documents are generated and archived over time throughout the project, files can be moved, modified and accessed by different individuals, without a consistent method of handling the files. Managers and others can benefit from audit logs that identify who has accessed particular documents, and when, in order to determine whether there are newer versions of the document, etc. Access history provides not only records of the file management and control details, but also some degree of security — helping users keep track of the history of activities in the archive. Existing tools, such as Buzzsaw, enable users to track and manage file versions by viewing activity logs. However, it is difficult for users to group the activities and visually get a picture of how the files had been accessed and modified in the history. Imagine the following scenarios. An architect is cleaning up a huge collection of design documents generated by the project team and the documents, which have not been used and updated for a long time, need to be moved to an obsolete folder. In another scenario, a project manager is preparing supporting documents of an issue found last November for discussion in the next team meeting; s/he wants to narrow the search results by extracting only the files that were accessed and modified during last November. Existing tools do not support these needs efficiently, especially when the individual needs to browse the access history of multiple documents. Therefore, adding the capability to visually represent the access history of a file would be an asset, helping users visually identify and group the access history more quickly.

Other than the conventional information retrieval and results representation, I aimed to provide users, through my prototype, with innovative and intuitive ways to search and access project archives. Since project archives and the access history contain historical information generated throughout the length of the project, organizing the search results and access history using a timeline-based layout became one of my key design ideas. (I will justify and discuss the benefit to individuals in the case studies and discussion chapters.) Visualization techniques allow information search and exploration tasks to be performed visually, which not only provides users with better awareness of information but also improves the process and experience while a user is interacting with the data. Users value time and efficiency, and because visual representations communicate information more quickly than can text, visualization techniques can help support efficiency goals. However, complex visualization techniques may not be a good solution in this case since most of the users in this industry are not information visualization professionals. Therefore, I focused on designing a simple visual representation, which most users can easily understand, to enable easier access to relevant information.

Based on the results of requirements gathering, I present my design objectives:

1) *Support relevance-ranked searching in historical archives and provide effective visualizations of the search results:* Provide relevance-ranking mechanism to generate search results with different levels of relevance to the search keywords, and filters that remove unnecessary information; provide interactive visualizations and supporting visual cues to visualize the project archive and search results, to help users distinguish different search results. These features allow users to visually find different relevant information and prioritize the information to view.
2) *Support flexible browsing, exploring and accessing of the archives:* Provide usable components such as multi-scale displays, scroll bars, and visual timeline displays to enable users to explore and interact with the data more easily.

3) *Support visualization of archive access history:* Provide users the visual capacity to view the access log of particular files as additional information in order to track actions undertaken by others.

At the current stage, I designed my prototype with requirements gathered from the construction domain. However, because most of these problems and requirements are common to other domains as well, I believe my design objectives can also be applied to other domains with appropriate modifications and customizations. For example, filters can be customized based on requirements and information from different domains. My design aims to provide users a combination of usable, visual, and interactive components to support better searching, browsing and exploring of information spaces.

3.2 Prototype Design

My prototype focused on integrating a combination of standard visualizations and interaction techniques to solve the specific problem of searching file archives. In this section, I describe the design details of my prototype, focusing on an overview of the interface, the interactive timeline visualization, the information browser, filters and an access history viewer.

3.2.1 Overview of the Prototype

VisArchive consists of the following visual and interactive components: search bar (Figure 1(a)), interactive Timelines (Figure 1(d)), information browser (Figure 1(b) and 1(c)), advanced filters (Figure 6), and access history viewer (Figure 7). Starting with the main user interface, the search bar is located at the top of the screen, in which users are able to start performing search tasks by typing in multiple keywords to search. Clicking the button on the right side of the search bar opens a popup window with advanced filters, which helps users narrow down the search results to be visualized and displayed. The information browser (Figure 1(b)) including description viewer (Figure 1(c)) allows individuals to browse the items within an archive and to view the meta-information and description of a selected item in detail. Two interactive timelines at the bottom of the interface (Figure 1(d)) visualize statistical information of the archive including one full-range timeline for the overall project archive and one scalable timeline for viewing a detailed portion of the file archive within a selected time interval.

Users can interact with the timelines by scrolling the time slider (Figure 1(e)) between the two timelines. The information shown in the timelines and information browser will be updated simultaneously while users are performing different search tasks and/or moving the time slider to view the archive in a different time range (Figure 1(f)). By performing a search task, search results will be generated behind the scenes by my relevance-ranking algorithm (described in section 3.3) and the relevance information related to the search keywords will be visualized in the timelines and information browser with additional visual representations to help users identify the most relevant search results and explore other related information in the file archive.

3.2.2 Interactive Timelines and Visualization of the Search Results

In the interactive timelines, time flows from left to right; when users have not yet performed a search task, the statistical information of the project archive is visualized initially as a grey colored bar chart in the timeline. The items in the archive are arranged in the timeline based on *creation time* by default. Each bar represents a particular time unit (e.g. one day in Figure 1) and the height represents the number of information items that have been created on that particular day.

The lower timeline provides the visualization of the overall project archive from the first day to the most recent day of the project. The bar chart over the timeline visualizes the statistical information overview of the project archive. The upper timeline provides the visualization of the project archive with a specific time range that is customizable by selecting the time interval from the dropdown list at the top right of the main interface (Figure 1(g)). The time interval options available to be chosen from the dropdown box are currently set as three days, seven days, fifteen days, one month, three months and six months. The upper timeline displays the same type of information as the lower timeline, but with different options to scale up the visualization. This provides users with the ability to view specific time ranges in more detail. The Y-axis of the timelines represents the count of the items within the specific date range and the X-axis represents the date of creation. The time slider (Figure 1(e)) between two timelines enables users to interact with the information displayed in the timelines and information browser. The light blue pane over the lower timeline indicates the current time range being displayed in the upper timeline. By moving the time slider horizontally, it updates the upper timeline visualization to a specific time range and the location of the light blue pane in the lower timeline simultaneously. As well, the information browser will be updated with the archive items located within the selected time range (described in Section 3.2.3).

To search the file archives, users input one or more keywords into the search bar located at the top. My prototype implements the concept of dynamic queries [11], which allows users to formulate search queries dynamically and get feedback immediately by adjusting the time slider in the timeline panel and clicking on the items in the information browser. The search results are assigned with different levels of relevance to the search query based on my relevance-ranking algorithm (described in Section 3.3). After that, the original grey bar charts in the timelines turn into color-coded stacked bar charts to represent the relevance-ranked search results (Figure 2 and 3). Figure 2 and Figure 3 show the timelines with the visualization of relevance-ranked search results while searching with two and four keywords, respectively. The timelines as shown in Figure 3 visualize all the files with different levels of relevance to the search keywords. The color-coded stack bar charts over the timelines show the statistics of these search results with different relevance levels throughout the life of the project.

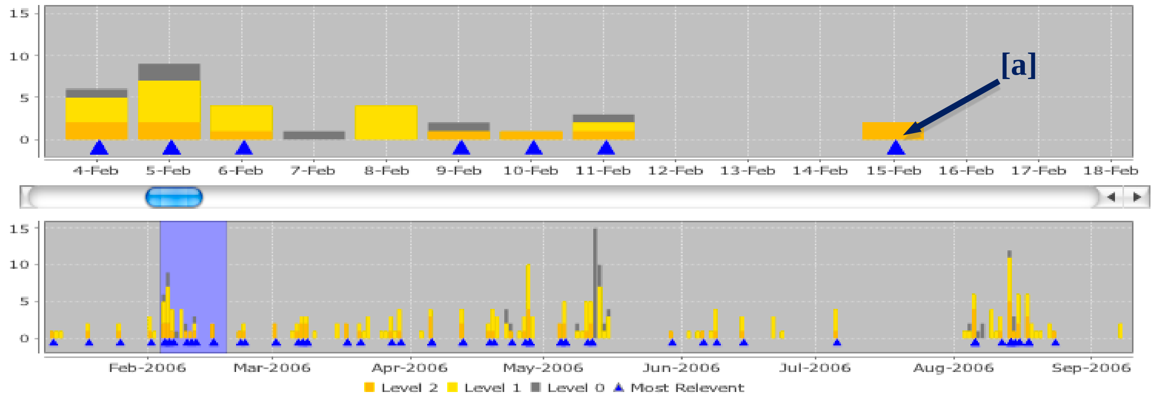

Figure 2. Timeline visualization while searching with two keywords (darker yellow indicates result with higher relevance level than lighter yellow).

(a)Blue arrow indicates the most relevant search results

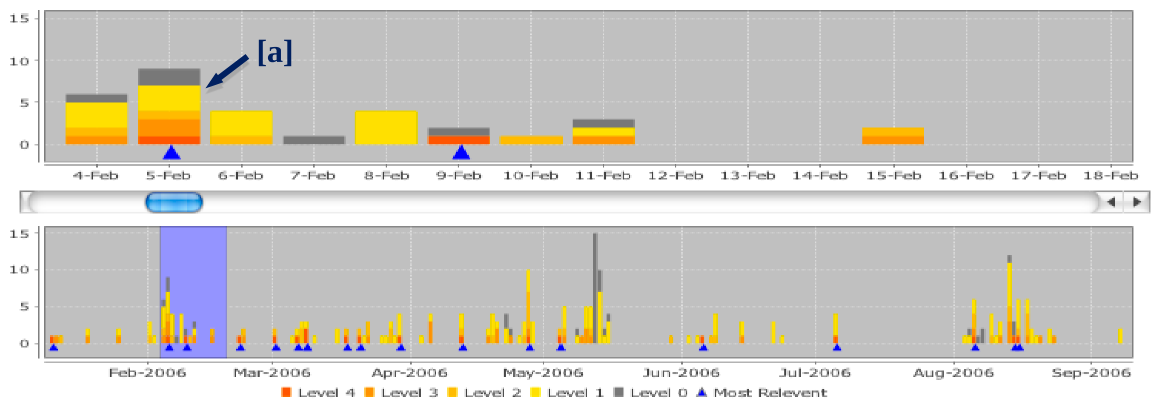

Figure 3. Timeline visualization while searching with four keywords.

(a) Stacked bar with color scale indicates the number of search results for each relevance level on that date

The levels of relevance for search results are represented by a color scale in my prototype. Each of the relevance levels from level 1 (least relevant) upwards will be assigned colors ranging from lighter yellow to dark red (Figure 4), respectively. As Figure 2 and 3 shown, more colors will be used as more levels of relevance are assigned to the search results. Grey color represents the archive items with zero level of relevance (i.e. none of the search keywords match the meta-data of the archive items). More relevance levels can be found in Figure 3 compared to Figure 2 since users are searching with more keywords. The darker color in the stacked bar chart represents the search results that are more relevant to the search keywords found on those dates. This color-coding is applied to the stacked bar charts in the timelines as well as in the information browser (described in Section 3.2.3).

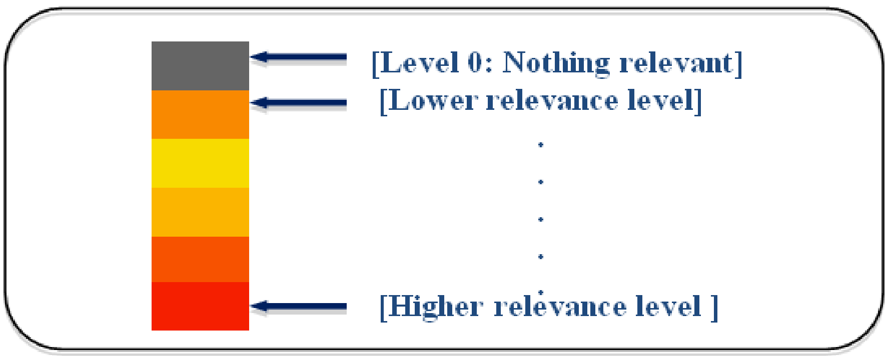

Figure 4. Color scale representing the levels of relevance

Blue arrows shown at the bottom of the bar charts in the timelines indicate that at least one of the search results in the particular dates matches all the search keywords and is considered to be one of the most relevant search results (Figure 2(a)). For example, in the upper timeline in Figure 3, the blue arrows indicate that the most relevant files were created on Feb 5th and Feb 9th, 2006. Therefore, users are able to identify the most relevant search results and their creation dates from either timeline using the visual cues of the blue arrows. The stacked bar charts with color scale in the timelines convey the relevance and quantitative information of the search results to the users. Users can visually identify both the search results with specific levels of relevance to the search keywords and the number of results on particular dates. The height of the color cell in the stacked bar represents the count of search results assigned with the relevance level. For example, the stacked bar chart for Feb 5th, 2009 (Figure 3(a)) indicates there are two most relevant results (dark orange), followed by other less relevant results in lighter colors. The color cells in the stacked bar are ordered in their relevance order from high to low, bottom to top, respectively (Figure 3(a)). Therefore, users can start looking for the search results with the highest relevance level from the bottom line of the stacked bar charts.

With the timelines, users are able to get an overview of the quantitative and relevance information of the search results. More importantly, the timeline visualizations create a picture of the relevance-ranked search results across the overall project. This conveys how information items relevant to the search keywords are distributed along the timeline. The time slider between timelines allows users to interact with the information details such as information ID, name, and summary, in the information browser. It enables users to browse and navigate to the items in the information browser for the specific time range based on the creation date.

3.2.3 Information Browser and Visual Cues Supporting the Search Results

Information items with relevance-ranked visual information will be updated and displayed synchronously in the information browser (Figure 5) as users adjust the time range in the timeline. The information browser lists the information items vertically and shows all the information items within the same time range that is selected in the timelines. In the example of the construction project archive, the file items are represented as rectangular boxes and associated with a file name in the box and file creation date on the right. Users are able to scroll through the information browser vertically to browse and select the file they would like to view. The items in the information browser are ordered by creation date. Upon the user clicking on a particular file item, the meta information including file name, file path and file description will be shown in the description viewer on the right side (Figure 5(d)). This information provides users a summary of the selected file item. Clicking the “View Access Info” button brings up a separate window to allow users to view the access history information of the selected file (described in section 3.2.5). Clicking the “View File” button opens the file on the desktop, through either a web browser or an appropriate software tool, allowing the user to view the full contents of the file.

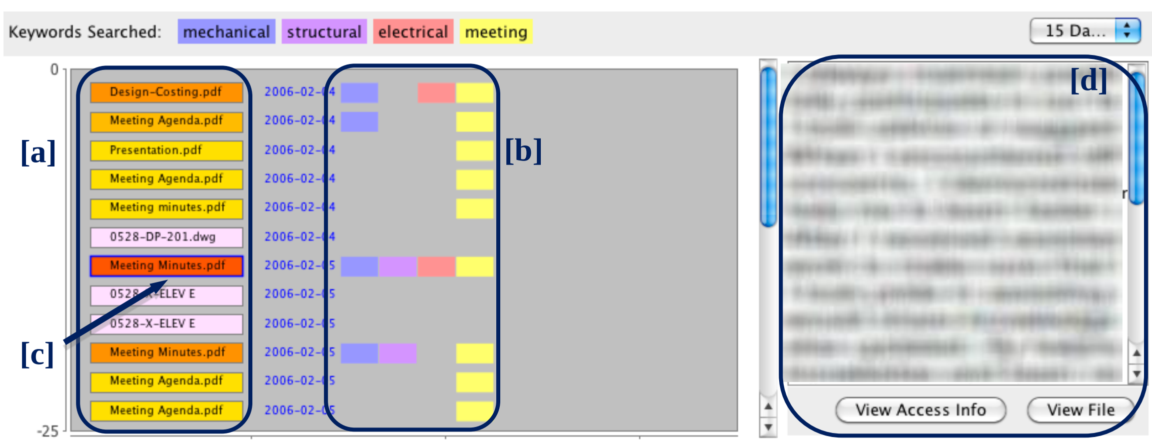

Figure 5. Information browser including description viewer displaying file items for a four keyword search.

(a) color-coded rectangles representing the search results’ associated relevance levels (b) color-coded panes identify the matching search keywords for each file (c) rectangle highlighted with blue border indicates the most relevant result that is matching all the search keywords (d) description viewer display the description of a file when users click the ticket of a file

For consistency, the color-coding used for visualizing the relevance-ranked search results in the timelines is used in the information browser to potentially help users identify the relevant items more easily and effectively. Figure 5 shows sample search results with four keywords. The rectangles (Figure 5(a)) representing information items are filled with scaled-colors to indicate the level of relevance to the search keywords. Instead of using blue arrows in the timeline, I use a blue border surrounding the scaled-color rectangles in the information browser to indicate the most relevant search results (Figure 5(c)). Therefore, users should be able to identify the most relevant search results very easily from the timeline as well as in the information browser. Moreover, the rectangles representing information items with scaled-colors allow users to explore other relevant file items in the archive with different relevance levels matching the search keywords.

The other important feature provided by my interface is that it provides the ability to distinguish search results matching different search keywords very easily and effectively. My prototype applied techniques similar to visual brushing and linking [41] to establish relationships and to distinguish between each group of data and provide *focus + context* information with multi-scale timeline views to support archive search tasks, respectively. In VisArchive, searched keywords are coloured with randomly assigned distinct colors and linked to each of the search results in the information browser when users perform a search task (Figure 5(b)). The color panes on the right side of each item represent the keywords that the item is matching. Since the order and color of the panes represent the search keywords, users are able to perceive visually the search keywords that match the search results instead of reading textual details to identify a match.

Therefore, users should only need to focus on browsing the results and reading the file names, details, etc. For example, in Figure 5(b), there is one file item matching all the search keywords “mechanical”, “structural”, “electrical” and “meeting”, and three less relevant items matching the keywords “mechanical” and “meeting”. Since the matched keywords have no relationship between one another, using colors can help user identify similarities and link to the matched keywords with greater ease. Uses can also simply ignore the items with no colored highlights as they are not relevant items to the search. This supporting visual cue (color-coded panes for each item) becomes very useful when users want to be able to distinguish between search results with the same level of relevance, but matching different search keywords, especially when one or more of the search keywords are prioritized over others. For example, in Figure 5, users might be more interested in exploring the search results relevant to “electrical meeting” than others. Therefore, by scanning through the information browser, although there are items with same relevance level, users can easily identify the item relevant to “electrical meeting” and ignore other items with the same — or even higher — relevance level according to their search priorities. This supporting visual cue allows users to distinguish the items easily and should enable users to explore the relevance details of the search results in the archive more effectively.

3.2.4 Advanced Filters

Filters shown in Figure 6 were designed based on the user requirements gathered from the construction project. The use of filters helps users narrow down the search results based on file contents and properties. In the construction project archive example, VisArchive allows users to filter search results by file types (Figure 6(a)), created users (Figure 6(b)), and keyword exception (Figure 6(c)). For example, a manager might want to see all the PDF and DOC files created by the team members that he is managing. The "Keyword exception" filter allows users to exclude the keywords that they are not interested in. For example, a user might want to search all the files related to floor plans but he is not interested in the "second level" floor plan. The idea of using filters is to allow users to limit their search by eliminating irrelevant and uninteresting items. By applying the filters onto the search tasks, the irrelevant items will not be processed by my relevance-ranking algorithm, and will not be visualized in either the timelines or the information browser.

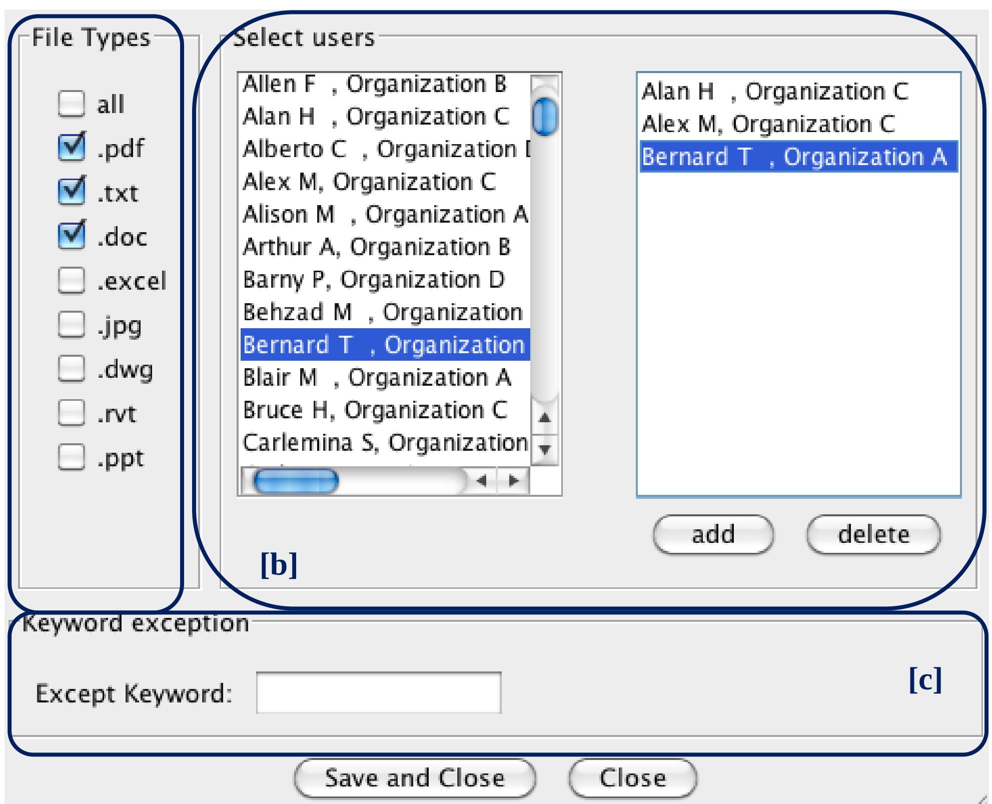

Figure 6. Advanced filters designed for the construction file archives

(a) File types; (b) User who created the file; (c) Keyword exception

In general, custom filters should be developed for each domain in order to conform to the information in the project archive and the searching preferences of users. In different domains, individuals involved in the project might be interested in different fields to be filtered when they are searching the information. For example, construction project individuals might be interested in searching files in the archive for specific file types, creation users, etc. Software developers might be interested in searching software defects for specific software components, release versions, etc. Filters are a supporting feature in my prototype but not a new contribution of the work.

3.2.5 Access History Visualization Viewer

Users can view access history information of a selected file item through a pop-up window by clicking on the button below the description viewer in the main screen. In order to minimize the learning curve, the design of the visualization and interaction for this panel is similar to the design of the VisArchive main screen, with similar visual representations and interactions. The access history visualization viewer (Figure 7) consists of a timeline visualization (Figure 7(a)) to visualize summary information about the access history, an access history browser (Figure 7(b)) to display the details of access history, and a user filter (Figure 7(c)) to filter the access history by access user name. The upper part of the access history visualization viewer visually displays the access records in the access history browser on the left side. Normally, people can distinguish objects with a small number of different colors easily, and thus colour becomes a useful signifier in revealing similarities between distinct item types within groups of information. The access history visualization viewer uses distinct colors to indicate different types of access visually, so that users can recognize how a file was accessed and how. In my prototype, green represents file creation, blue represents file access, and red represents file modification. However, the color encoding used in the access history viewer could be easily adjusted to adapt to different needs (e.g., use different colors to accommodate red / green colorblind users).

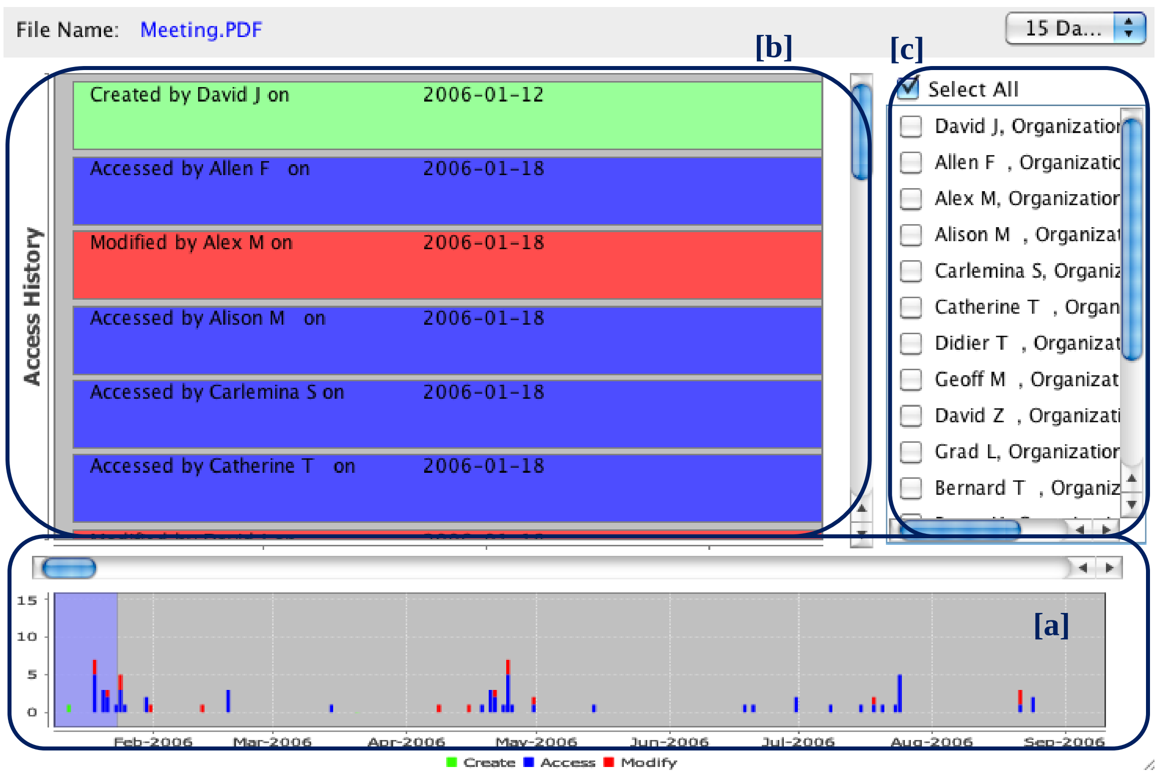

Figure 7. Visualization showing access history of a selected file
(a) Timeline visualization; (b) Access history viewer; (c) User filter

The timeline visualization of the access history located at the bottom allows users to get an overview of all type of access made to the file since the file was created. Based on the access date, the timeline visualizes the count of accesses by using a stacked bar chart and the type of each access by using distinct colors shown in the stacked bars. To view the detail, a user can scroll the time slider to an interesting date range and browse the detail of access information from the access history browser. The information detail includes the date of access and name of the accessing person. The colors used in the access history browser represent the type of access, and are identical to the color used in the timeline panel. This visual support was designed to enable users to distinguish visually the type of access more quickly than reading textual information.

3.3 Algorithm for Generating and Visualizing Search Results

My research was not focused on how users pick the keywords for searching. Instead, my interface is designed to minimize users' search effort and reduce the need to try different combinations of search terms randomly. The relevance-ranked search results are generated by my relevance algorithm and represented visually on the interactive timelines and information browser by applying visualization representations and supporting visual cues. The goal of my relevance algorithm is not to create the best algorithm for generating search results in the work. Instead, I aimed to demonstrate the idea of integrating relevance-ranked search results with a visual representation to enable users to visually search and explore the archives more easily and intuitively. My relevance algorithm could be easily replaced by any other ranking algorithm if different relevance criteria were desired.

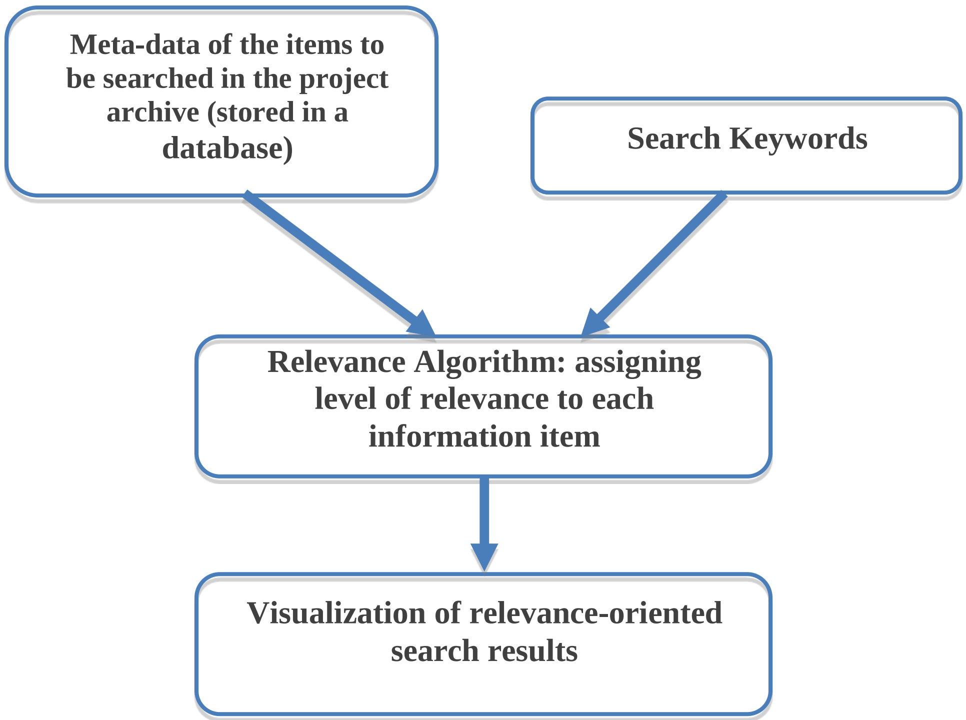

Figure 8. Diagram of generating relevance-ranked oriented search results

To generate the relevance-ranked search results, the algorithm calculates a relevance ranking based on the search terms and assigns the ranking to each information item in the project archive. As I described in Figure 8, my prototype first extracts the meta-information of each item from the project archive database (e.g. the meta-data of the files in the construction project archive contains filename, file path, file keywords and description). The algorithm then matches this extracted information with the search keywords input by the user to compute the relevance levels for each item. At the end, the prototype prepares the search results with the assigned level of relevance for data visualization that is presented to the users. Since the accuracy of generating the search results before visualization is not my focus at this stage, my algorithm is simple in order to demonstrate the idea of visualizing search results in terms of levels of relevance.

Higher relevance level will be assigned if the meta-data of the item matches more search keywords. The level of relevance will be increased by 1 if any one of the search keywords is found in the meta-data of the item regardless of the number of the times that keyword appeared. Level 0 will be assigned if none of the search keywords is matched. For example, I assign the level 0 of relevance to the items if none of the searched keywords was found in the extracted meta-data of the item. I assign level 5 to the item if 5 of the search keywords were matched. Therefore, every time users input keywords to perform a search task, all the items in the archive will be assigned levels of relevance from 0 up to the number of search keywords. As I only used a sample of files and data from the archives for the case studies, the system generated the relevance-ranked search results in less than a second. The result set was then processed with visualization techniques and visually represented to users in the user interface. Note that for very large archives, adaptations may be needed, such as to the relevancy algorithm and application user interface. My current prototype can support archives with thousands of files without system performance issues.

3.4 Implementation

VisArchive was implemented as a desktop application using Java and the JFreeChart toolkit [33]. Most of the charting and visualization used in my prototype were generated by using the JFreeChart API, with modifications and customizations. My prototype requires a database to store the project archive as information records, file access history and/or a central file repository to store the electronic files of the archive if digital files are part of the project archive. In order to make the archive content searchable, I needed to extract textual information as meta-data for keyword-based searching from the electronic files. For ease and efficiency of generating a dataset to demonstrate the concept of my interface, I extracted and created this information manually from a subset of the existing archives. From the construction project archive, I indexed the electronic files by extracting all necessary meta-information regarding each document and integrating this information into the database for demonstration purposes. The meta-data that I extracted from the files consisted of file name, file description, date of creation, related keywords and file path. File access history data was stored separately in a different table from file meta-data in the database.

VisArchive is a front-end desktop client that communicates with the database and file repository and generates the search results to support the archive search and data visualization. The data to be visualized and used by my prototype are stored as entries in a database. Since the construction file archives were stored as electronic files in a central file repository, a file parser could be developed in the future to extract the meta-data from the file and parse this information into the database automatically. The repository management system may allow users to tag related keywords as meta-data to a file manually when they create or modify the files. The details and limitations of meta-data extraction will be discussed in Chapter 5.

Chapter 4: Case Studies

In this chapter, I examine the feasibility of VisArchive for searching, browsing and exploring information in project archives of different domains by demonstrating two case studies: (1) a construction project and (2) an open source software project (software defect tracking). The case studies aim to demonstrate my prototype and examine how well it could achieve my design objectives of supporting search and exploration tasks by (1) organizing the project archive and search results using a timeline-based layout, (2) visually representing search relevance and which search keywords were matched, and (3) visually representing the file access history. I focus on my prototype's capacity to resolve complex use scenarios, rather than simple use scenarios like searching files with known file names or IDs, which can normally be done easily without the support of visualization.

The interface of VisArchive has been revised and modified based on these case studies, but the core features have remained stable in order to support understanding of the value the three design ideas provide for supporting search tasks within each domain. Since a file contains more information than users often need, the interface for the construction project case study was designed to include a description viewer that allows users to view the details (e.g. file description or file path,) when they click on a file from the information browser. Separately, users can identify a software defect item by viewing its summary. I designed the interface for the software project case study to display the summary for each software defect in the information browser.

4.1 Case Study: Construction Project File Archives

In this case study, I use VisArchive to search a file archive of the construction project that I discussed in Chapter Three. The construction project was to design an educational building for a university. The project archive contained more than 800 files that were created by different individuals involved in the project, including project managers, project coordinators, engineers and architects. I obtained about 300 files based on their available information to construct my file archive for piloting my prototype. These files were stored and shared as digital copies in a central hosting server with a variety of file types such as PDF, DOC and TXT. Since this project file archive was not shared with the public, I could not display some of the private and confidential information from the archive such as names of people involved and detailed contents of the files. Detailed contents of the file have thus been modified in the design sections and case study to protect names, personal and proprietary information.

For demonstrating VisArchive in this case study, I extracted and imported into the archive information such as ID, name, path, and description of each file, and then archived this information as searchable meta-data into a database system for archive searching, data processing and visualization by VisArchive as described in Chapter Three. I assumed that users would be able to determine whether the file was what they were looking for by viewing the extracted meta-data. For this case study in particular, I demonstrated my access history visualization component for visualizing file access history in the archive. Since prior file access history data was not provided by the construction file archive, I created a synthetic file access history for a number of files in the database for demonstration purposes.

I was not interested in searching known files with known information (e.g. file name or ID) because this type of search can be completed easily, without exploring the archive. Instead, I focused on searching and exploring the files in the file archive that users might have never accessed before, or the files that were known by users, but for which users would likely not remember specific information to help them find the file, such as the file's name or ID.

4.1.1 Searching Files that Match all the Search Keywords

From results of the early study in the construction domain [37], I learned that project individuals frequently need to access and review project files such as meeting minutes, agendas and design documents, through the duration of the project in order to keep the project on time, on budget and to the high level of quality required of infrastructure. They described scenarios where they were searching the files in the archive without success — sometimes files that they had accessed before, but for which they had forgotten the exact name or ID, and at other times, for files that they believed should be a part of the project, but that they knew may not exist in the archive at all. For example, a project manager shared a scenario in which he wanted to find all "electrical", "mechanical" and "structural" documents that an engineer, whom I will refer to as "Mike," was involved in evaluating. Mike had left the company and the project manager needed to identify all files he had been working on in order to share them with Mike's replacement. With traditional search methods provided by existing tools such as Buzzsaw, search results are presented as a list of files from top to bottom with file name, size, last-modified date, etc. Although current search solutions may enable the most relevant files to appear at the top of the list, the project manager would not know clearly which keywords and how many of them were matched. Accordingly, the manager might need to open each file to evaluate how relevant it is to the search keywords. It would also be difficult for the project manager to understand how these files had been produced through the project duration. This is important because the manager might only want to find files that were produced in a certain period in the project's history.

Using my prototype, the project manager should be able to identify the files he needs more easily by glancing at the blue arrow indicators in the lower timeline (Figure 9). Here, the manager can easily identify that there are files matching all the search keywords ("electrical," "mechanical," "structural," and "Mike") on 10 different dates. These files are considered the most relevant to the manager's search. Although these documents may or may not be relevant to all of the building's components (electrical, mechanical, structural), these files must contain *all* of the search keywords entered and the name "Mike" in order to show up in the search results. Thus VisArchive would help the project manager to view the most relevant files first — more quickly than would be possible when searching manually. Most importantly, VisArchive makes it clear which files matched all search terms, and when not all search terms are matched, it is clear which ones. Some search terms may be more important to the query than others.

This timesaving can be critical for finding most relevant information in a large construction project archive, because time constraints are usually an issue during these projects. In the VisArchive, the blue arrows in the timelines not only answer the question about whether there are any relevant files in the archive — matching all the search keywords — but also provide users a visual overview of when these files were created during the construction project. In a situation like the one described above, where a new engineer was taking over for another engineer, project individuals might want to focus a document search on the files created within a specific construction stage (e.g. during the period of Mike's employment in the scenario described previously). For example, if the project manager is seeking files relating to the construction of the foundation of the building, which he believes was completed before March and he knows that Mike worked on the project between January and September, the project manager may choose to look specifically for files created in February. From the timeline shown in Figure 9, between Feb 2nd and Mar 2nd, two files appear to be most relevant (Feb 5th and Feb 21st). The project manager could then easily navigate to those files by interacting with the timeline. To choose the best file of the relevant results, users are offered more information about each file in the information browser, which allows users to view and access detailed information about the files (Figure 10).

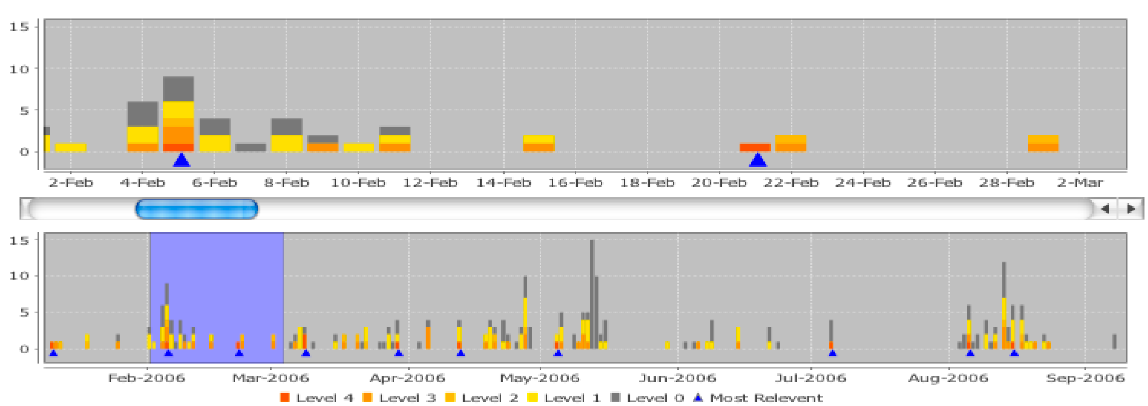

Figure 9. Timeline visualization while searching keywords: electrical, mechanical, structural, and Mike

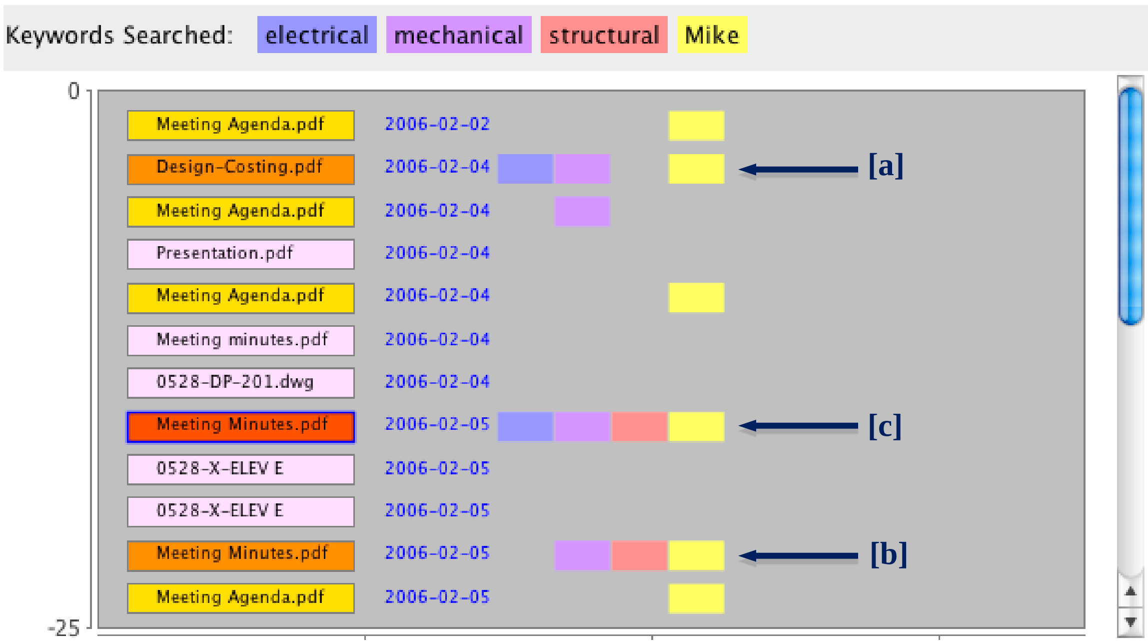

Figure 10. Information browser displaying the files with color-coded visual supports. (c) Most relevant file; (a and b) Second most relevant files

4.1.2 Exploring Files relevant to the Search Keywords

In VisArchive, the color-coded stacked bar charts in the timelines and visual support in the information browser are designed to enable users to explore files with different levels of relevance to the search keywords. Besides finding the files that match all the search keywords ("electrical," "mechanical," "structural," and "Mike"), the manager might be also interested in exploring other files relevant to one or more of these keywords considered to be less relevant. For example, he may want to find other "electrical" related files which Mike was involved in, or find other files related to "structural" and "mechanical" — files not necessarily containing all three keywords. Existing search solutions for construction project archives make it difficult for individuals to explore the files by their relevance to the search keywords. Since there is no visualization of the search results, all files are returned in a conventional list. Individuals must view the textual meta-information of a file to identify its creation date and matched keywords. While the file list can be arranged by either "Date" or "Relevance", individuals cannot easily explore and browse the relevant files in the project timeline arranged in both "Date" and "Relevance". Therefore, it becomes difficult to answer the questions that make search more efficient. E.g. are there any files that match three of the four search keywords? When were these files created in the project timeline? Which month contains more relevant files than the others? With VisArchive, however, the color-scaled visual support in both the timeline and the information browser should allow users to identify the relevance-ranked files more effectively and efficiently. For example, the upper timeline in Figure 9 shows that there are files relevant to three of four search keywords (level 3) on Feb 5th, Feb 9th, Feb 11th, Feb 15th, Feb 22nd and Mar 1st respectively. Although these are not the most relevant files, as they do not match all four of the search keywords, they are still highly relevant and thus recommended to the user for consideration as well. Using VisArchive then, users can identify the level of relevance for each file easily in the information browser. In my example, two files (Figure 10(a) and 10(b)) are shown as less relevant than the most relevant file (Figure 10(c)), but highlighted as more relevant than the other files found in the search results.

The associated color-coded visual panes for search keywords in the information browser allow users to distinguish the files with same relevance but different matched keywords effectively. For example, when a manager wants to explore electrical documents with which Mike was involved (files containing "electrical", and "Mike"), other files relevant to other search keywords might also be shown with the same relevance (e.g. files containing “structural” and “Mike”). With the existing solutions, users need to read extra meta-information of each file in order to differentiate between files with the same relevance level. With VisArchive, in the information browser (Figure 10), the color panes representing matched search keywords of each file visually differentiate these files, helping the user to distinguish the most relevant documents. For example, in Figure 10, blue, purple, red and yellow represent the keyword “electrical”, “mechanical”, “structural”, and “Mike”, respectively. The color panes show that one file (Figure 10(a)) shown in the information browser matches the keywords “electrical”, “mechanical” and “Mike”, another file (Figure 10(b)) matches the keywords “mechanical”, “structural” and “Mike”. Although both files are assigned with the same relevance level, and indicated with the same color scale in the timeline and the information browser, the manager can easily identify that the file shown in Figure 10(a) is the file containing “electrical” and “Mike”.

Besides searching and exploring files in the project archive, project individuals might want to explore other information of the project archive from the timeline visualization. For example, a manager might be interested in identifying the time periods in which the project archive was more active (i.e. when more files were created), and in such cases the relevant files in those time periods would be the most interesting for further exploration. In VisArchive, the color-coded stacked bar charts on the timelines show the density of file creation, and the density of files relevant to the search keywords throughout the life of the project. By viewing the timelines individuals can easily identify the dates or time periods in which more files were created and/or the periods in which more files were created that are relevant to the search results. For example, the timelines in Figure 10 show that more files were created in the month of February, May, and August. Therefore, if the manager was seeking files created during periods of more activity in the project archive — for example, if there had been a dedicated resource supporting document management during a certain period — then the manager might be interested in exploring the files that were created in those periods. As well, because the timelines convey information about the various activities and file types created during the project (e.g. documents such as blue print plans may have been created most frequently earlier on in the project) the manager can make informed decisions about which documents are most relevant. Thus, when an exact time period is not known, VisArchive can help individuals narrow down time periods in which to search by clearly displaying the periods in which activity in the project archive was at its highest or at its lowest.

4.1.3 Exploring File Access History

Users might be interested in the access history information of files in the archive for project management purposes (e.g., an architect has placed his latest version of the design plan into a shared archive and he wants to know whether the consultants have accessed it after he uploaded the new version to ensure that the they are working on the latest version; a manager might want to investigate whether a design document is the one that has been reviewed and updated by the team during the last redesign process.). The color-coded visualization of access history provides much of this information, such as the number of times the file was accessed and the type of access (e.g., opening or modifying a file), helping users to identify the desired file effectively. In the access history viewer, each type of access is assigned a color-code to improve the users' ability to identify the file they are looking for (e.g., the one they accessed the day before or the one that was modified most recently). In Figure 11, by viewing the access histories through the timeline, a user can see that the file has been accessed or modified in January, April and May. Moving the time slider on the timeline allows users to view the detailed access information of the specific time range in the browser above. For example, a user can see that the access history browser shows that the file "Meeting.pdf" was created on Jan 12th, 2006, accessed and modified on Jan 18th, 2006 by David.

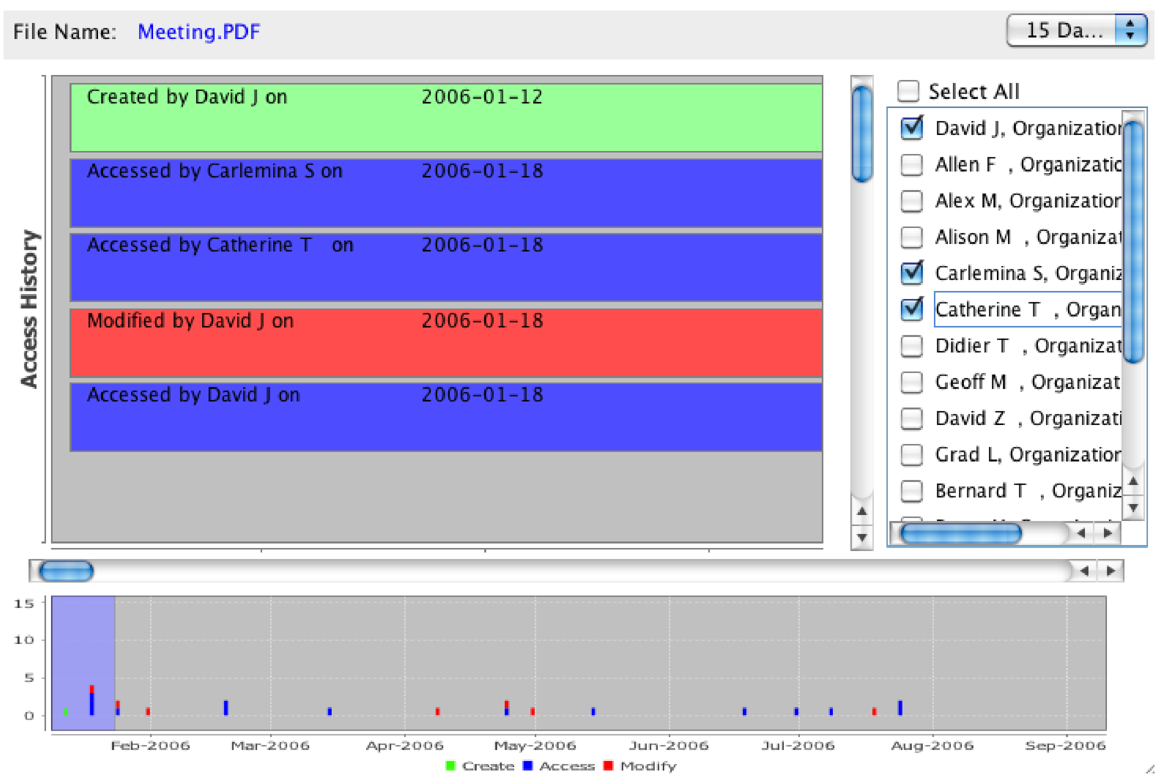

Figure 11. Visualization of access history filtered by selected access users

In addition, users are able to filter the access records by selecting the individuals from the filter (Figure 11) on the right. The filter lists all the individuals whom have created, accessed, or/and modified the file. Let’s imagine a manager wants to know when David, Carlemina and Catherine have accessed or modified the file. In Figure 11, by selecting the first three names (David, Carlemina, Catherine) from the list, the access records related to the selected individuals are displayed on the screen and others’ irrelevant access histories are filtered out of the search results.

4.2 Case Study: Defects Tracking of Mozilla Thunderbird Project

In a second case study, I expanded the application of VisArchive beyond the construction file system into a software defects tracking system in the open-source software development domain. Different from the construction project file archives, software defects were not structured into directories as digital files, and they were only stored virtually in a centralized database management system, so users could create or access the defect information by using a software client through the Internet. I expect VisArchive to provide project contributors more thorough information for searching and exploring the defect database. Therefore, relevant defects can be found more easily and quickly and the chance of logging duplicated defects can also be reduced.

The Mozilla project was started in 1998 and was intended to develop open-source software project with the power of thousands of programmers all over the world [34]. Thunderbird is the Mozilla Foundation’s next-generation email client. As the software is being used all over the world by thousands of users, software defects and issues can be found and reported by using a web-based defect tracking tool called Bugzilla [35] that also allows developers to track these issues and, eventually, fix them. The Thunderbird project archived more than 5000 software defects in Bugzilla from the beginning of the project in 2004, which enabled users, in this case study, to focus on searching and visualizing larger amounts of data than was possible in the construction project case study. I exported around 1000 defect records from Bugzilla under Thunderbird project between 2006 and 2007. For each software defect, I obtained the defect ID, date, and summary, as meta-data for use by VisArchive.

As was the case with the construction project archives, finding information over thousands of defect records in Bugzilla can be a tedious process. Bugzilla allows users to search the defect archives by entering keywords and using advanced filters that are similar to the search mechanism of VisArchive. However, the search results are represented in a conventional list of defect information (Figure 12). Although users can reorder search results alphabetically by attributes, it is difficult for users to view the relationships among different defects, and especially, to explore defects that are partially relevant to the search keywords. In this case study, I modified the information browser of the prototype as a simplified varient of VisArchive that is different from the version I used for the construction case study.

1240 bugs found.

ID ▲	Status ▲	Opened	Summary
199374	UNCO	2003-03-26	--disable-composer breaks mailnews
201153	UNCO	2003-04-08	Entering a comma in the TO field splits names in addresbook
312324	UNCO	2005-10-13	Going online and sending mail, opening 'unsent messages' in new window sends message twice (new mailnews window opened interpreted as a 'going online' event)
342367	UNCO	2006-06-21	Sending a message hangs saving to Sent folder
597726	UNCO	2010-09-18	Imported mail has incorrect dates and contains HTML tags
598742	UNCO	2010-09-22	Big-5 gets set on replies
601047	UNCO	2010-09-30	No memorized drop-down list for multiple addresses in a single Bcc: line
606743	UNCO	2010-10-23	Add Support for Control-D on New Message
647789	UNCO	2011-04-05	Warning to avoid accidental placement of large address books in the To: and Cc: fields
648233	UNCO	2011-04-07	Using quote shortcut (ctrl + shift + .) in reply to email often crashes Eudora OSE
655540	UNCO	Sat 16:30	Send multiple addressees from Address Book
147898	UNCO	2002-05-29	joining/splitting lines doesn't work correctly
174630	UNCO	2002-10-15	Easier way to file sent mail
220123	UNCO	2003-09-23	Request "Send and File" option/button on Compose window to copy message to folders other than "Sent"
224474	UNCO	2003-11-02	For message sent to mail recipient and newsgroup (Reply-to-all to a post) want to use different identities for SMTP and NNTP

Figure 12. Conventional list of search results provided by Bugzilla

The defect summary was described by the defect finder, and contained the crucial information needed by a software developer or tester to recreate the defect. This kind of meta-data is hard to categorize and filter using the original Bugzilla interface. Therefore, I focused on finding relevant software defects by searching and exploring through the summary of software defects.

4.2.1 Searching Defects in the Software Defect Archives

In order to fix issues and improve the quality of software, developers need to search and find the software defects from the archives that correspond to their expertise or the components they are responsible for. In addition, time information, such as when the issues were filed, is also useful for developers to prioritize the issues to fix. Figure 13 shows an example of search results and visualization support that is provided by

VisArchive. By glancing at the timeline visualization of search results, the software developer should be able to easily identify whether the summary of any software defect contains all the search keywords (“message,” “compose,” and “window,”) and the date that these defects were created. In this example, there are eight days indicated by the blue arrows in the timelines that contain the most relevant results. From here, the developer might be interested in finding the defects that have been in the archive longer, which cannot be easily seen in the conventional list of defect results provided by Bugzilla. With VisArchive, the developer can navigate to the time range containing the earliest and most relevant defect found (Figure 13(a)) in the timeline by using the time slider, and view the defect summary of the most relevant defect (Figure 13(b)) in the information browser. Regardless of the size of the archive, the blue arrows can always provide awareness of the most relevant results and should enable the users to find these results effectively and efficiently.

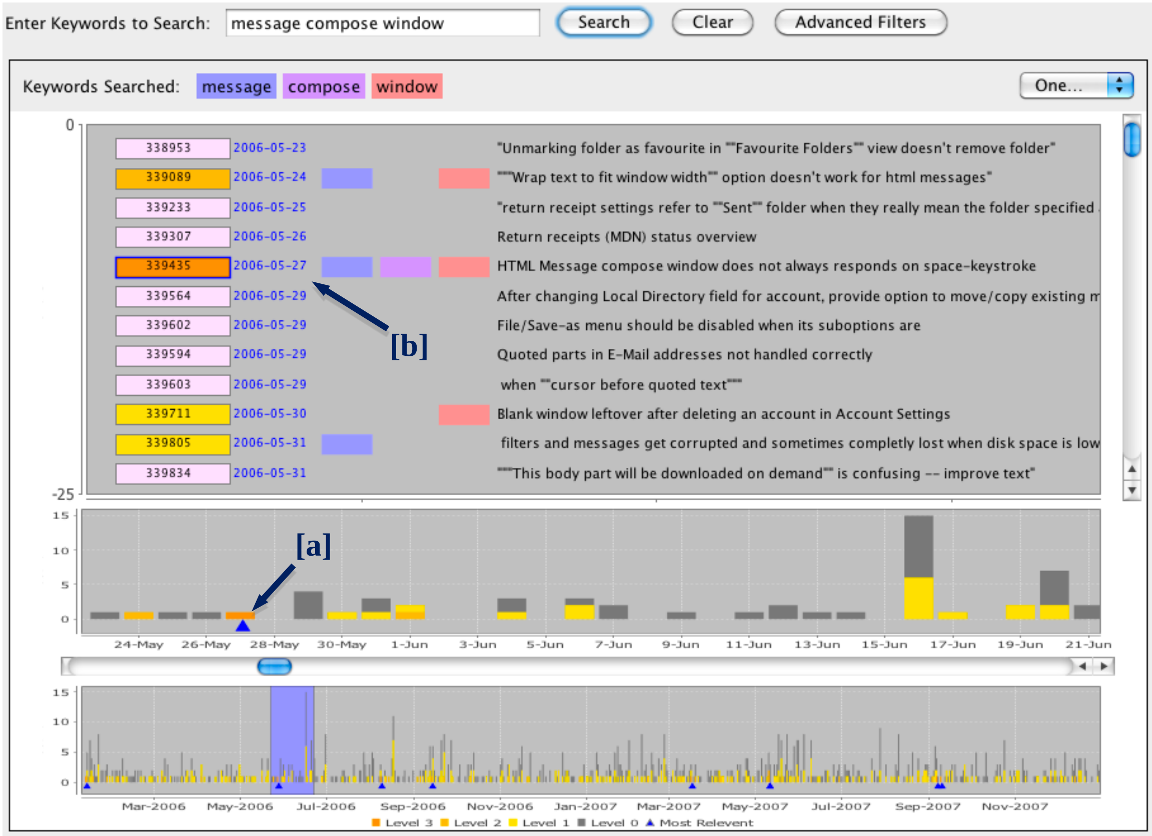

Figure 13. Visualization of search results provided for searching "message compose window" in the software defect archive

(a) Most relevant defect indicated in the zoomed timeline; (b) Most relevant defect indicated in the information browser

4.2.2 Exploring the Defect Archive and Relevant Software Defects

If none of the defects matches all the keywords or users want to explore other software defects with less relevance, users may refer to the stacked bar charts in the timeline and visual panes to perceive how the defects in the archive are relevant to the search keywords and where these defects occur on the archive timelines.

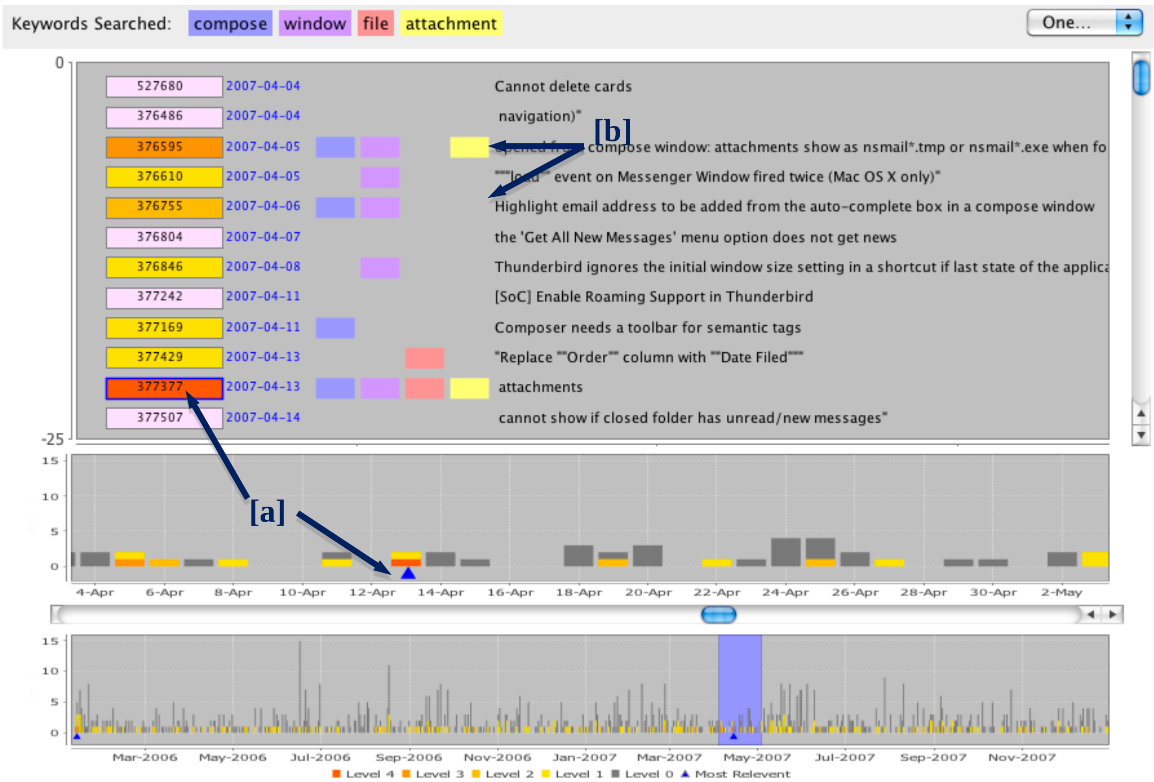

Figure 14. Visualization of search results provided for searching "compose window file attachment" in the software defect archive

(a) The most relevant defect; (b) defects matching "compose window"

Figure 14 shows the example in which a software developer searches the software defect archive with more keywords than the example in 4.2.1 (e.g. searching "compose," "window," "file" and "attachment"). The developer can easily identify the results (Figure 14(a)) matching all the search keywords (e.g. defects might be about "file attachment" in "compose window") by finding the blue arrow in the timeline and the blue highlighted tickets in the defect browser. The developer may also want to explore other defects that are partially relevant to the search keywords. For example, the developer may be interested in other defects relevant to "compose window" or "file attachment" (e.g. Email "compose window" might have other issues besides in the "file attachment" function, and the developer may want to fix those as well). Other keyword combinations (e.g. "compose attachment", "window file") are not the terms that the developer is concerned with in this case, and thus these software defects can be ignored. By glancing at the stacked bar chart over the timeline, the developer can easily perceive how the defects in the archive are relevant to the search keywords and how these defects distribute over the timeline in the archive. The developer can browse and explore the defects that partially match keywords by visually scanning the color panes of keywords in the defect browser, instead of reading the summary of defects (e.g. the defects containing "compose window" are interesting items (Figure 14(b) and 15(a)), whereas the defects containing "window file" may be disregarded (Figure 15(b))).

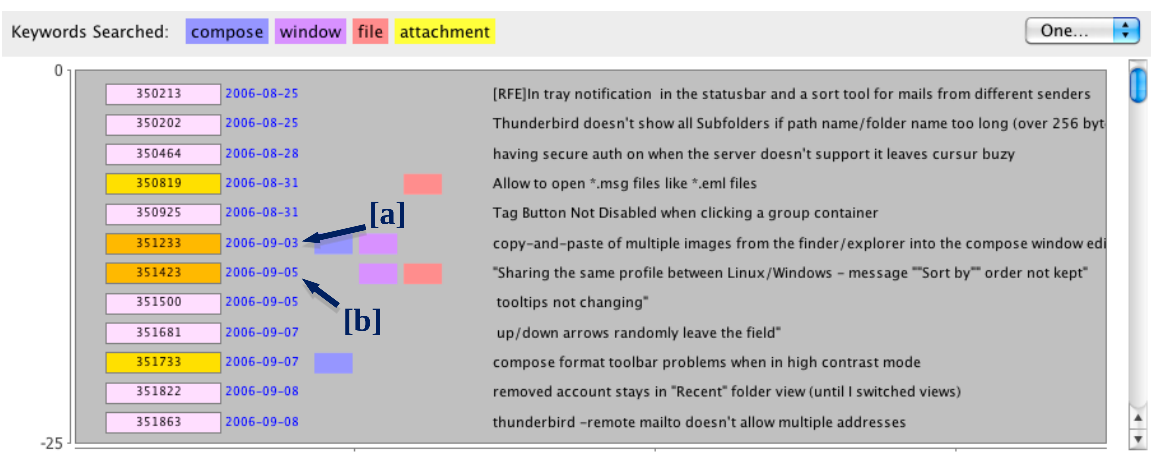

Figure 15. Sample information displayed in information browser
(a) the defect matching "compose window" (b) the defect matching "window file"

Similar to the construction case study, the timelines of the defect archive show a picture that conveys to the developer how many defects and how the defects in the archive match the search keywords. The developer is able to determine — visually — the dates that contain the defects that are more or less relevant to the search keywords. When used for logging new software issues, VisArchive enables software testers to search and explore whether there are similar or related issues existing in the archive. If relevant defects matching the same keywords are found in the archive, timelines enable users to determine and explore their relationship over time. As an example, a tester searches for a system bug in which a "Removed account stays in 'recent' folder view," even after it has been removed. The search keywords for this bug include “folder” and “preferences,” and the results show the bug was logged in August. The timeline visualization reveals that another defect, labeled, "Unmarking folder as favourite in 'Favourite Folders' view doesn't remove folder," is also associated with the keywords of "favourites," "folder" and "preferences.” The tester sees the bug was logged in May and has since been resolved. This information can thus allow the developer to explore whether the resolution to the earlier bug logged in May could suggest potential resolutions for the similar bug detected in August.

Chapter 5: Discussion

VisArchive enables users to effectively and rapidly find and manipulate information in historical project archives including exploring access history of the information. Instead of providing novel visualization and/or new interaction techniques to address information retrieval problems, I designed a new interface as a combination of multiple usable techniques that should make the tool easily understood and usable by even a novice user. As mentioned at the outset, my primary research goal was to develop a new interface approach that would enable people to search, explore and access relevant information from the archive more effectively and efficiently. I tried to solve the information retrieval problems of temporal project archives in an innovative and intuitive way. In my case studies, I demonstrated how VisArchive supports users to search, browse and explore archives with two case studies in different domains. In this section, I review and evaluate the three core design ideas of my VisArchive prototype, specifically:

1. Timeline-based interface for showing search results.
2. Visual indication of search relevance and which search keywords are matched.
3. Visual representation of file access.

5.1 Timeline based interface for showing search results

Again, my intent with VisArchive was not to create the best search algorithm — and certainly, any other ranking algorithm could replace mine if different relevance criteria were desired. Instead, I sought to employ visual representation of the relevance-ranked search results to offer visual cues to users that help them cognitively distill the information to complete their search more intuitively, and possibly identify and explore relevant files more effectively and efficiently. Search experience and interaction with the search results with my prototype are different from conventional search interfaces such as web search and Buzzsaw because VisArchive provides an interaction experience that enables users to browse, navigate and explore the archives while minimizing the steps of browsing the information. In a conventional search results list, search results are represented from top to bottom, page to page, and relevance rankings from higher to lower. If a user's search keywords are consistent with what he is searching for, the top few returned items might be the results that the user wants, thus a conventional representation of search results may be very easy to navigate. However, in many cases, including those described earlier in my prototype design and case study sections, further exploration is needed in order to evaluate more comprehensive results in historical archives. VisArchive visualizes the search results using timelines. The temporal information and quantitative information provided in the timeline visualization enables users to discover more information from the archive. It helps users prioritize the search results to review based on factors such as the age of the information and the density of nearby relevant results. Because VisArchive offers a timeline slider for navigating relevance-based search results, it eliminates the need to jump from page to page, enabling the user to slide the timeline to browse and navigate through results in the archives. Though this approach could increase efficiency for exploratory search, especially in my case studies, it is not without limits. For example, when a project is very long-term (such as a project over ten years) and the user cannot narrow down the time frame in which the files being searched for were created, the user will likely not experience much lift in efficiency. This is because scrolling through such a long timeline would be nearly as cumbersome as clicking from page to page through search results. Additionally, when moving from page to page, a person can easily track where they are in the search process by remembering the page where they left off, if, for example, they are interrupted while performing a search. With a visual timeline, however, the fluidity makes it more difficult to recall exactly where one last looked when interrupted.

5.2 Visual indication of search relevance and which search keywords are matched

The color-coded visual panes for search keywords in the information browser of VisArchive should allow users to distinguish between files with the same relevance, but different matched keywords, more effectively by indicating the matched search keywords of each file through visual cues that help users identify files most relevant to their needs. As we saw in the construction project example, the visual support of color-coding means users can avoid the need reading extra meta-data of each file in order to differentiate between files with the same relevance level. The visualization and visual supports provide better understanding of the search results that allow users to seek and find relevant files visually rather than reading the detailed context of the information in the archive. In the bug tracking system example, a blue arrow in the timeline with blue highlighting in the defect browser indicated the results that matched all the search keywords the user entered. When a user also wanted to explore defects that were partially relevant, the stacked bar chart over the timeline offered the level of relevance for each defect item in the archive and showed where each falls on the timeline. The visual supports (blue arrows in timeline and rectangle highlighted with blue border in information browser) allow users to visually identify the most relevant search results (results that match all the search keywords). The timeline with visualization of relevance-ranked search results allows users to explore different levels of search results relevant to the search keywords. The visual panes associated with each item in the information browser enable users to distinguish the items with different matched keywords effectively. The color-coding can be adjusted according to users' preferences or needs; for example, someone who has a colour vision deficiency can adjust to color indicators they are able to perceive.

5.3 Visual representation of file access

In my prototype design and case study sections, I described various scenarios in which people would need to view the access history of files, and explained how this information could be useful for decision making in a project. Since access information is time-oriented and can be grouped (e.g., accessed, modified), it can benefit from the timeline interface and visual cue design provided by my prototype. I also compared my prototype to traditional methods of viewing access information and explained how my prototype could help for exploring and understanding the access information. As I showed in the construction project example, the color-coded visualization of access history on the timeline interface provided more thorough information about a file's history such as the number of times it had been modified and accessed. This coding can help the user identify the desired file more efficiently. More importantly, it provided a picture of the full access history of the file on the timeline, which offered users a better understanding of how the file had been accessed; it does this in one holistic view. I anticipate that this approach will save users time in browsing, counting, and finding access information in the history (e.g., an architect could easily ensure that the design plan he updated had been accessed by the team members before the meeting; a manager could recall if a document is the one used in the design process 6 months ago). To better justify the usability of this feature, user studies need to be performed in the future.

5.4 General Discussion

VisArchive combines existing visualization and user interaction techniques in a novel way to support searching, browsing, and exploring information in historical project archives. I characterized the problems in project archive exploration and came up with a combination of design ideas to solve these problems. These represent my main contributions. While the visual cues for both the historical context and the matching-level of searches should make search efficient for users, the cues will only be successful if the users notice and understand them. To this end, more research is needed to confirm that users can indeed intuitively understand what the visual indicators mean in VisArchive in order to benefit to ensure users benefit from these features. Moreover, The design of VisArchive still needs further validation in terms of effectiveness, efficiency and comparison to other existing tools via user studies. I believe that my prototype can be applied as long as the information items in the archive have temporal information such as creation date and meta-data associated with them. Therefore, only the access information (e.g. creation date, access date, and modification date) and searchable meta-data (e.g. summary, description, keywords, tags) are required by my prototype regardless of the domain content. For example, research papers in the IEEE papers online archives [36] are associated with a publication date and meta-data for users to search. Instead of presenting the search results in a conventional list of papers, VisArchive could visualize a timeline-based overview of the published papers. I also expect that the advanced filters and information browser could be customized and modified based on users' needs in different domains.

My current prototype used manually extracted meta-data for demonstration purposes. However, my key design ideas of VisArchive should be able to seamlessly integrate with existing archive management systems that provide well-designed content management and text search capabilities. For example, the defect tracking system Bugzilla allows users to create and edit defects with searchable content and meta-data such as related keywords. Since all the content and meta-data have been stored in database when the defect was created, they became searchable by the system. VisArchive could be added to the system in place of the conventional text based search results. It would provide visualization and interaction capabilities to support searching, browsing and exploring the archive. However, in a file or document based archive such as the construction archive, searching can be more challenging in real life. In order to enable the search capability, searchable data such as textual content or meta-data needs to be extracted from the file or inserted by users when creating them (this is especially important if the file is not text based such as image files). Well-designed archive management systems should be able to extract the text content automatically from the text based files and make them searchable in the system. More importantly, they could allow or even require users to insert related keywords for better archiving the files, especially image based files. The quality of the search results can tremendously depend on the searchable meta-data. Therefore, how well the system can extract the content from the files, and how well users can classify the files through manual tagging, both will affect the quality of the search algorithm and usefulness of the visualization.

5.5 Improving on the VisArchive prototype

My current prototype focuses on supporting users to search and explore time-based (temporal) information archives. Therefore, it might be applied appropriately to other time-based project archives. However, I do think that VisArchive can be improved in many aspects to better support searching, browsing and exploring archives. The following are some suggestions and potential improvements:

(1) Interactive Timeline Visualization: each bar of the bar chart over the timeline represents the number of items created in one day. However, if the project archive covers a long period of time, the bar chart will be compacted (e.g. the lower timeline of the software defect archive, visualizing around one thousand records over two years), so the user will not be able to see the color-coded visualization clearly in the lower timeline. Therefore, an option to set the unit for bar chart representation would be useful (e.g. the unit of a bar can be set to represent a day, a week or a month). Currently, my prototype arranges and visualizes the data over the timeline by using creation date. It will be useful to enable users to arrange the data by different types of date such as last modification date.

(2) Relevance-ranking algorithm for generating search results: My algorithm to generate search results is simply keyword-based. However, I might apply different algorithms for searching archives in different domains because appropriate search algorithms for different domains might improve the quality and reliability for generating the relevance ranking for search results.

(3) Advanced Filters: filters should be customizable for different domains.

(4) Exploring Access history: currently my prototype only allows users to view the access history of a single item (e.g. a file or a defect record) in the project archive. It will be useful to enable users to explore the access history of multiple items in the archives. Different visual representations might be used to achieve this goal in the future research (e.g. I could use multiple timelines to represent the access history of different files in one display, or I could aggregate them into one timeline and use color-scale to distinguish different items. However, I need to consider factors such as the number of items to view at one time.

Chapter 6: Conclusion and Future Work

I presented a novel interactive visualization tool called VisArchive that integrates multiple commonly used visualization techniques to facilitate searching, browsing, and exploring historical project archives. VisArchive visualizes relevance-ranked search results statistically with color-coded stacked bar charts in project timelines and uses additional supporting visual cues to distinguish search results based on search keywords, which should provide project participants with better understanding of the search results. To show the utility of VisArchive, I presented two case studies as sample applications, and I discussed how VisArchive could be modified and enhanced to support other applications in different domains.

I mentioned the limitations of my prototype and explored the aspects that could be improved upon to provide better use capabilities and user experience. In the future, I recommend conducting user studies to examine the effectiveness, usability and user experience of VisArchive in various domains. By collecting the results from user studies, VisArchive could be enhanced with improved functionality, features, and user experience to better support searching, browsing and exploring historical project archives.

Bibliography

1. Card, Stuart K., Jock D. Mackinlay, and Ben Schneiderman, eds. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

2. Krishnan, Aparna, and Steve Jones. "TimeSpace: activity-based temporal visualisation of personal information spaces." *Personal and Ubiquitous Computing* 9.1 (2005): 46-65.

3. Bates, Marcia J. "The design of browsing and berrypicking techniques for the online search interface." *Online Information Review* 13.5 (1989): 407-424.

4. Vassileva, Julita, and Carl Gutwin. "Exploring blog archives with interactive visualization." *Proceedings of the working conference on Advanced visual interfaces*. ACM, 2008.

5. Plaisant, Catherine, et al. "LifeLines: visualizing personal histories."*Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1996.

6. Kumar, Vijay, Richard Furuta, and Robert B. Allen. "Metadata visualization for digital libraries: interactive timeline editing and review." *Proceedings of the third ACM conference on Digital libraries*. ACM, 1998.

7. Allen, R. B. "Interactive Timelines as Information systems interfaces", *Symposium on Digital Libraries*, Japan, August 1995, pp. 175-180.

8. Ahlberg, Christopher, and Ben Shneiderman. "Visual information seeking: tight coupling of dynamic query filters with starfield displays." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1994.

9. Padia, Kalpesh, Yasmin AlNoamany, and Michele C. Weigle. "Visualizing digital collections at archive-it." *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*. ACM, 2012.

10. Wu, Fuqu, and Melanie Tory. "PhotoScope: visualizing spatiotemporal coverage of photos for construction management." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009.

11. Ahlberg, Christopher, Christopher Williamson, and Ben Shneiderman. "Dynamic queries for information exploration: An implementation and evaluation."*Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1992.

12. Shneiderman, Ben, et al. "Visualizing digital library search results with categorical and hierarchical axes." *Proceedings of the fifth ACM conference on Digital libraries*. ACM, 2000.

13. Foo, Schubert, and Douglas Hendry. "Desktop search engine visualisation and evaluation." *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*. Springer Berlin Heidelberg, 2007. 372-382.

14. Sebrechts, Marc M., et al. "Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.

15. Veerasamy, Aravindan, and Russell Heikes. "Effectiveness of a graphical display of retrieval results." *ACM SIGIR Forum*. Vol. 31. No. SI. ACM, 1997.

16. Silva, Nabani N., et al. "Visual exploration of large collections in digital libraries." *Proceedings of the Latin American conference on Human-computer interaction*. ACM, 2003.

17. Mackinlay, Jock D., George G. Robertson, and Stuart K. Card. "The perspective wall: Detail and context smoothly integrated." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1991.

18. Richter, H., et al. *A Multi-Scale Timeline Slider for Stream Visualization and Control, GVU Center, Georgia Institute of Technology*. Technical Report GIT-GVU-99-30, 1999.

19. Dork, Marian, et al. "Visgets: Coordinated visualizations for web-based information exploration and discovery." *Visualization and Computer Graphics, IEEE Transactions on* 14.6 (2008): 1205-1212.

20. Shen, Rao, et al. "Exploring digital libraries: integrating browsing, searching, and visualization." *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2006.

21. Wongsuphasawat, Krist, and Ben Shneiderman. "Finding comparable temporal categorical records: A similarity measure with an interactive visualization."*Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, 2009.

22. Fails, Jerry Alan, et al. "A visual interface for multivariate temporal data: Finding patterns of events across multiple histories." *Visual Analytics Science And Technology, 2006 IEEE Symposium On*. IEEE, 2006.

23. Plaisant, Catherine, et al. "LifeLines: using visualization to enhance navigation and analysis of patient records." *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 1998.

24. André, Paul, et al. "Continuum: designing timelines for hierarchies, relationships and scale." *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007.

25. Goren-Bar, Dina, et al. "KNAVE II: the definition and implementation of an intelligent tool for visualization and exploration of time-oriented clinical data."*Proceedings of the working conference on Advanced visual interfaces*. ACM, 2004.

26. Gilbert, Eric, and Karrie Karahalios. "Lifesource: two cvs visualizations." *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2006.

27. Viégas, Fernanda B., Scott Golder, and Judith Donath. "Visualizing email content: portraying relationships from conversational histories." *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006.

28. Gilbert, Eric, and Karrie Karahalios. "CodeSaw: A social visualization of distributed software development." *Human-Computer Interaction–INTERACT 2007*. Springer Berlin Heidelberg, 2007. 303-316.

29. Froehlich, Jon, and Paul Dourish. "Unifying artifacts and activities in a visual tool for distributed software development teams." *Proceedings of the 26th International Conference on Software Engineering*. IEEE Computer Society, 2004.

30. Jones, Simon, et al. "Visualization of Heterogeneous Text Data in Collaborative Engineering Projects."

31. Walk, Simon, et al. "PragmatiX: An Interactive Tool for Visualizing the Creation Process Behind Collaboratively Engineered Ontologies." *International Journal on Semantic Web and Information Systems (IJSWIS)* 9.1 (2013): 45-78.

32. Autodesk Buzzsaw: http://www.autodesk.com/products/buzzsaw/ [Accessed April 1, 2014]

33. JFreeChart Visualization Toolkits: http://www.jfree.org/jfreechart/ [Accessed April 1, 2014]

34. Mozilla Project: http://www.mozilla.org/ [Accessed April 1, 2014]

35. Bugzilla: https://bugzilla.mozilla.org/ [Accessed April 1, 2014]

36. IEEE: http://www.ieee.org/index.html [Accessed April 1, 2014]

37. Tory, Melanie, et al. "Physical and digital artifact-mediated coordination in building design." *Computer Supported Cooperative Work (CSCW)* 17.4 (2008): 311-351.

38. Isenberg, Petra, and Danyel Fisher. "Collaborative Brushing and Linking for Co-located Visual Analytics of Document Collections." *Computer Graphics Forum*. Vol. 28. No. 3. Blackwell Publishing Ltd, 2009.

39. Pirolli, Peter, and Stuart Card. "Information foraging in information access environments." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995.

40. Pirolli, Peter, Stuart K. Card, and Mija M. Van Der Wege. "Visual information foraging in a focus+ context visualization." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001.

41. Buja, Andreas, et al. "Interactive data visualization using focusing and linking."*Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*. IEEE, 1991.