Hu et al. Visualization in Engineering (2016) 4:6
DOI 10.1186/s40327-016-0036-8

Visualization in Engineering
a SpringerOpen Journal

RESEARCH Open Access

VisArchive: a time and relevance based visual interface for searching, browsing, and exploring project archives

Keyun Hu, Sheryl Staub-French*, Melanie Tory and Madhav Prasad Nepal

Abstract

Background: Project archives are becoming increasingly large and complex. On construction projects in particular, 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.

Methods: This research investigates a query-driven approach that represents new forms of contextual information to help users understand the set of documents resulting from queries of construction project archives. Specifically, this research extends query-driven interface research by representing three types of contextual information: (1) the temporal context is represented in the form of a timeline to show when each document was created; (2) the search-relevance context shows exactly which of the entered keywords matched each document; and (3) the usage context shows which project participants have accessed or modified a file.

Results: We implemented and tested these ideas within a prototype query-driven interface we call VisArchive. VisArchive employs a combination of multi-scale and multi-dimensional timelines, color-coded stacked bar charts, additional supporting visual cues and filters to support searching and exploring historical project archives. The timeline-based interface integrates three interactive timelines as focus + context visualizations.

Conclusions: The feasibility of using these visual design principles is tested in two types of project archives: searching construction project archives of an educational building project and tracking of software defects in the Mozilla Thunderbird project. These case studies demonstrate the applicability, usefulness and generality of the design principles implemented.

Keywords: Visual analytics, Access history, Relevance-ranked search, Visualization, Construction project, Software defects

Background

Electronic data storage and database management systems offer simple and inexpensive ways to store digital information and documents of a construction project and enable project team members or software tools the ease and capacity to access project information remotely from anywhere. Construction documents such as meeting agendas, meeting minutes, schematic diagrams and computer-aided design (CAD) drawings, cost data, project schedule, design specifications contain rich information about different facets of a construction project. This information is typically archived in a shared digital storage repository. Unfortunately, even though this rich 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. In order for such repositories to be of practical use, construction professionals need to be able to rapidly retrieve and manipulate relevant information from the large and diverse collection of documents within project archives (Steed et al. 2012; Strotgen and Gertz 2012; Tory et al. 2008).

* Correspondence: ssf@civil.ubc.ca
University of British Columbia, Vancouver, Canada

Several recent studies, however, have shown that finding the right documents from a project archive in a timely manner can be very difficult. A 'project archive', in the context of this research is defined as a collection of files or information being generated or recorded historically throughout the project and stored in a common shared repository. For example, on several of the construction projects we have studied, design and construction teams used Autodesk Buzzsaw® (Buzzsaw 2014), a third-party application that is often used as a central repository for project information archiving, sharing and retrieval. Tory et al. (2008) found that design and construction professionals using these existing tools had a difficult time searching and locating project files unless they were already familiar with the hierarchy structure and the name of the item they were searching for. Similarly, Demian and Fruchter (2006a) reported that construction professionals often asked colleagues for information rather than searching project archives. They argue that project archives may provide insufficient contextual information to help people understand the meaning of retrieved documents.

Documents in construction project archives are typically organized and stored in hierarchical directories similar to other types of project archives. This allows individuals to access files by browsing directories and searching files using the meta-data (keywords, date, authors, etc.). Demian and Balatsoukas (2012) argue that there are two unique issues in the design of information retrieval systems for the construction domain: (1) Engineers and construction professionals are unique in terms of their information needs and information-seeking behavior, and (2) construction project archives are organized differently than other types of document collections. Consequently, they conclude that there are unique design challenges for construction project archives and a need for research into the design of query-driven systems for this domain.

Research into the design of such query-driven systems has shown the importance of revealing *contextual information* about the results of a user's textual query rather than simply a relevance-ranked list of resulting documents (Demian and Balatsoukas 2012; Demian and Fruchter 2006b). Specifically, users need to see resulting items in the *hierarchical context* of their ancestors, descendants, and siblings within the file hierarchy, and understand the granularity of the item within that hierarchy (Demian and Balatsoukas 2012). Our work extends query-driven interface research by providing two new forms of contextual information to help users understand the set of documents resulting from their query: 1) *temporal context*, in the form of a timeline to show when each document was created and 2) *search-relevance context*, by showing exactly which of the entered keywords matched each document. In addition, we provide a new approach to reveal *usage context* for a particular item, so a user can explore which other users have accessed or modified a file. We introduce these ideas within a prototype query-driven interface, *VisArchive*.

VisArchive employs visualization techniques to reveal the various types of contextual information, thereby offloading cognitive effort onto the perceptual system (Card et al. 1999). It employs a combination of multiscale and multidimensional timelines, color-coded stacked bar chart, additional supporting visual cues and filters to support searching and exploring historical project archives. The timeline-based interface integrates three interactive timelines as *focus + context* visualizations (Steed et al. 2012; Pirolli et al. 2001). It implements and extends *Dynamic Queries* (Ahlberg et al. 1992) and *Visual Information Seeking* principles (Ahlberg and Shneiderman 1994). The feasibility of using these visual design principles is tested in searching construction file archives of an educational building project. We also apply our tool to a software defects tracking system in the open-source software development domain to demonstrate its ability to search and visualize larger amounts of unstructured data.

Motivation, Methods, and Needs

This section describes the unique challenges of working with construction project archives and the specific user needs for visualizing archived construction project information. The visualization design espoused in this section is primarily motivated by a common problem encountered in the construction domain. However, the concepts could also be applied to other domains that have non-spatial, metadata-based and time-oriented project data, such as source files of a software project.

Construction projects generate voluminous data sets with significant heterogeneity of data files (structured, semi-structured and unstructured) (Russell et al. 2009) and types (Knowledge and Information Management 2006; Rezgui 2001). Electronic documents for a construction project are archived and stored over time in a central repository which we refer to as a 'project archive'. The files can be moved, modified and accessed by different individuals from diverse backgrounds and from different organizations. Design and construction practitioners frequently search for relevant project information within the project archives on a regular basis as part of their intra- or inter-organizational decision making processes. As such, the ability to search, browse and explore project archives more easily and effectively is critical for the success of a construction project. Specifically, construction experts need an effective and efficient way to find and classify information (Caldas et al. 2002) and, more importantly, to explore relevant information in the project archive (Demian and Balatsoukas 2012).

Previous research (Demian and Balatsoukas 2012) and our own studies of construction projects (Tory *et al.* 2008)) have demonstrated that information seeking and retrieval from large, shared construction project archives is a very difficult and time-consuming process. In particular, these studies found that it is difficult for project members to search for and locate relevant documents, and that there is a lack of support to browse and explore search results in construction project archives. Practitioners can spend significant amounts of time searching for documents and in many cases, fail to find what they are looking for. For example, in one meeting we observed, the mechanical consultant spent over 10 min searching for images of the water filtration systems on his laptop, which significantly interrupted workflow in the meeting. In another meeting, the project team was unable to find the sustainability goals for the project and ended up spending significant amounts of time trying to identify the goals from memory and writing them out by hand on a white board.

The digital files of the construction project we studied were archived into different directories organized in a hierarchy and stored in a central shared repository using Autodesk *Buzzsaw*® (Fig. 1). Project files could be accessed by browsing the hierarchical directories and project members had some flexibility in saving the digital files in different directories or in their own designated space. Project participants could view the audit logs to identify who and when other members of the team accessed particular documents. Although these tools enabled users to track and manage file versions, it was extremely difficult for users to group the activities and visually get a clear picture of how the files had been accessed and modified. It was also challenging to explore the project archive and search for information that project participants were not familiar with. These challenges impacted meeting productivity, disrupted group discussion, and impeded decision-making, all of which can be costly to a construction project.

The challenges associated with searching, browsing, and exploring construction project archives illustrates the important *contextual information* needed to help users understand the set of documents resulting from their queries. Specifically, we identified the following contextual information that informed the design requirements we considered in the development of *VisArchive*:

- Search Relevance Context: Relevance-ranked searching of project archives and effective visualization of search results is critical for project participants to: (i) generate search results with different levels of relevance to the search keywords and filter unnecessary information; (ii) provide interactive visualizations and supporting visual cues to

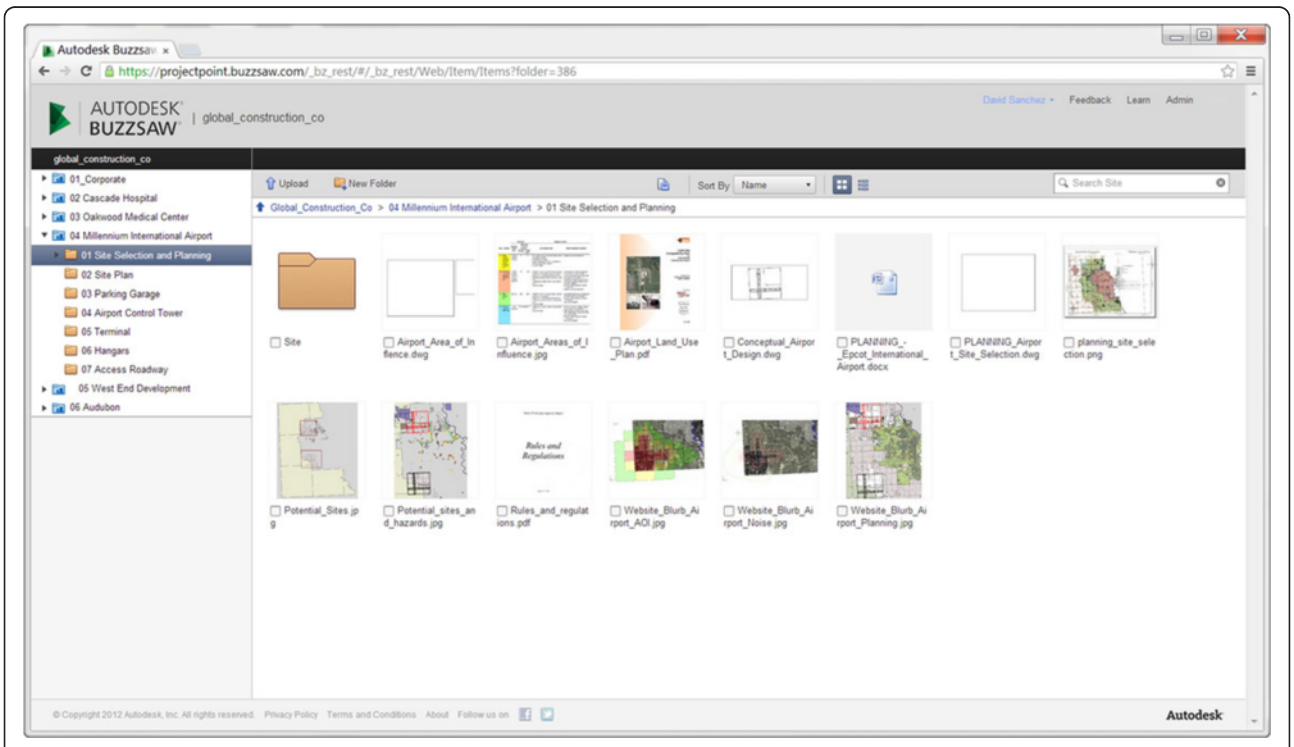

Fig. 1 Snapshot of Autodesk Buzzsaw document repository (Source: www.autodesk.com)

visualize the project archive and search results; and (iii) help users distinguish the different search results. *VisArchive* implements features that allow users to visually find relevant information and prioritize the information to view.

- Temporal Context: Practitioners need the ability to browse, explore and access project archives for specific time periods or across different time horizons. *VisArchive* provides usable components such as visual timeline displays, multi-scale displays, and scroll bars so that users can easily explore and interact with the project archive.
- Usage Context: Users need the visual capacity to view the access log of particular files as additional information in order to track actions undertaken by others and to visualize archive access history.

VisArchive represents this contextual information using a combination of usable, visual, and interactive components to better support searching, browsing and exploring project archives. While the design of the current prototype is based on the requirements gathered from the construction domain, we believe that most of these problems and requirements are common to other domains as well.

Relevant visualization techniques

As a visualization-based interface, *VisArchive* utilizes and extends various visualization design ideas. In this section, we summarize existing design contributions related to *VisArchive*'s three context-specific design features: timeline based visualizations; visual indications of search relevance and matched keywords; and visual representations of user activity in shared repositories.

Timeline-based visualizations

Timelines have been widely used in variety of applications and domains to visualize and present historical and temporal data. They provide aids in uncovering important relationships of searched results or entities, cues for filtering of information, and assistance to identify specific patterns of search results (Kwon et al. 2012). They've been used to explore temporal metadata and relationships in digital libraries (Kumar et al. 1998), visually browse and explore a blog archive by using a time slider (Indratmo et al. 2008), visualize email archive content (Viegas et al. 2006), visualize distributed software development consisting of code repositories and project communication (Gilbert and Karahalios 2007), explore temporal patterns of events within medical histories (Fails et al. 2006), visualize digital collections of web resources (Padia et al. 2012) and visualize design process with evolving building information data (Kim et al. 2011), among other examples. Timelines can also be used as an interactive filter for information indexed by time (e.g., Wu and Tory 2009; Jones et al. 2013).

More relevant to us are search and query interfaces that provide some temporal context to the search results. Previous research in information retrieval has shown that enabling a user to see temporal context of their search results and to sort or filter by time can be very useful (Alonso et al. 2009, Dumais et al. 2003). Commercial search tools (e.g., Google Scholar) also make use of temporal context interfaces. However, much of this previous research has been done in a general internet search context and has focused on extracting useful temporal information from the documents to support temporal clustering and queries (e.g., Alonso et al. 2009, Dumais et al. 2003, Hoffart et al. 2011, Jones and Diaz 2007). Temporal feedback interfaces in these systems are typically limited to a simple facet where a user can filter a textual results list by entering or selecting a time range, plus the ability to sort results by time. Jones and Diaz (2007) do present a timeline visualization to complement the textual results list, but at a very abstract level showing only key temporal clusters, not actual documents. We suspect that temporal context may be even more important for information seeking in project archive interfaces because participants in a project may be able to recall the approximate date when an item was created or used.

Visual indication of search relevance and matched keywords

Various techniques have been developed to reveal the relevance of retrieved documents to search keywords. Veerasamy and Heikes (1997)) designed a visualization that displayed the relevant documents and assisted users to effectively reformulate queries based on the searched keywords in the first stage. Foo and Hendry (2007) created and evaluated a suite of visualizations for searching one's desktop. 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, these visualizations were not space efficient. Neither of these tools integrated temporal context into the search process.

Cambiera (Isenberg and Fisher 2009), a tabletop visual analytics tool, supports information foraging activities in large text document collections. It allowed users to visually connect different groups of data or activities and updates in different visualizations for a visual analytics task, particularly using color-coded search keywords. However, *Cambiera* focused on providing awareness of users' analytical activities to others in a collaborative search task. *VisGets* (Dork et al. 2008) 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. (2013) 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. However, the visualization design process does not include interactive searching, browsing and retrieval of information.

Visual representation of user activity in shared repositories

Visualization of time-based human activities has a long history. For example, *Lifelines* (Plaisant et al. 1998) visualized medical records of a patient such as past symptoms, diagnoses and medications through an interactive timeline-based interface. Within the context of project repositories, the vast majority of this work has focused on software development repositories, with greater emphasis on changes to files and source code rather than activities of the developers (Storey et al. 2005). For example, *Augur* (Froehlich and Dourish 2004) visualized software artifacts and development activities with color-coded indications over the source code. Those projects that have focused on activities of developers have tended to support understanding the overall project evolution (e.g., Ogawa and Ma 2008), rather than information seeking tasks such as finding all files matching given criteria.

The *Timespace* (Krishnan and Jones 2005) 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). *PragmatiX* (Walk et al. 2013) 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. There is a limited previous work that focuses specifically on visualizing file access logs, which can assist users in searching and exploring temporal shared project archives.

Results: System Design and Implementation

This section describes the development of, *VisArchive*, a visualization tool that implements a combination of standard visualizations and interaction techniques to solve the specific problem of searching project archives. We first provide a detailed description of the visualizations implemented, and then describe the relevance algorithm and the implementation details.

Overview of the visualization tool

VisArchive consists of the following visual and interactive components: search bar (Fig. 2(a)), information browser (Fig. 2 (b) and (c)), interactive Timelines (Fig. 2 (d)), advanced filters and access history viewer (discussed in section "Evaluation"). The search bar allows users to input multiple keywords as well as the option for advanced filters. The information browser (Fig. 2 (b)) including description viewer (Fig. 2 (c)) allows users 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 (Fig. 2 (d)) visualize 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.

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 (Fig. 2(f)). By performing a search task, search results will be generated behind the scenes by relevance-ranking algorithm (described in "Relevance algorithm") 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.

Interactive timelines and visualization of the search results

The items in the archive, by default, are arranged in the timeline based on *creation time*. The time range selector provides the visualization of the project archive within a customizable time range or interval (Fig. 2(g)). Users search the file archives based on keyword/s. *VisArchive* implements the concept of dynamic queries (Ahlberg et al. 1992), which allows users to formulate search queries dynamically and get feedback immediately through manipulation of the time slider and information browser. The search results are assigned with different levels of relevance to the search query based on the relevance-ranking algorithm.

The levels of relevance for search results are represented by a color scale. 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). The continuum of lighter yellow to dark red indicates the increasing level of relevance. This color-coding is applied to the stacked bar charts in the timelines as well as in the information browser.

Blue arrows shown at the bottom of the bar charts in the timelines indicate the most relevant files created on particular dates that match all the search keywords and are considered to be one of the most relevant search

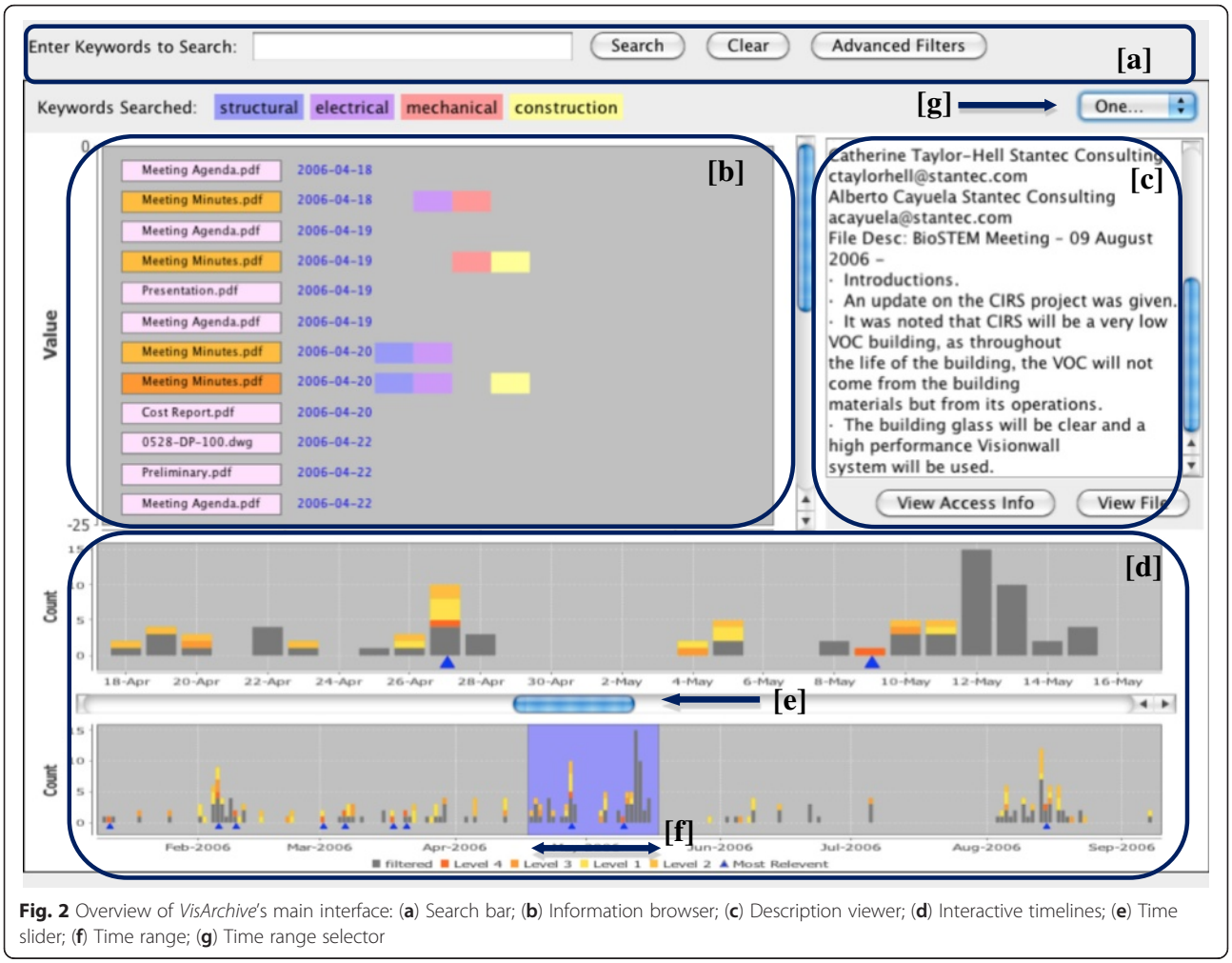

Fig. 2 Overview of *VisArchive*'s main interface: (**a**) Search bar; (**b**) Information browser; (**c**) Description viewer; (**d**) Interactive timelines; (**e**) Time slider; (**f**) Time range; (**g**) Time range selector

results. Users are thus able to identify the most relevant search results and their creation dates from the timeline using the visual cues of the blue arrows.

Information browser, visual cues and advanced filters supporting the search results

Information items with relevance-ranked visual information are updated and displayed synchronously in the information browser 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 (Fig. 2 (c)). Users are able to scroll, browse and select the file of their interest and identify its meta-information including the access history information of the selected 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. 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.

VisArchive allows users to distinguish search results matching different search keywords. We applied techniques similar to visual brushing and linking (Buja et al. 1991) to establish relationships and to distinguish between each group of data and provide *focused + contextual* information with multi-scale timeline views. In *VisArchive*, searched keywords are colored with randomly assigned distinct colors and linked to each of the search results in the information browser when users perform a search task (Fig. 2 (b)). The supporting visual cue (color-coded panes for each item) allows users to distinguish between search results and explore their relevance details in the archive.

The use of filters (Fig. 3) helps users narrow down the search results to be visualized and displayed based on file contents and properties such as by file types (Fig. 3 (a), created users (Fig. 3 (b), and keyword exception (Fig. 3(c)).

In general, custom filters should be developed for each domain in order to conform to the information in the

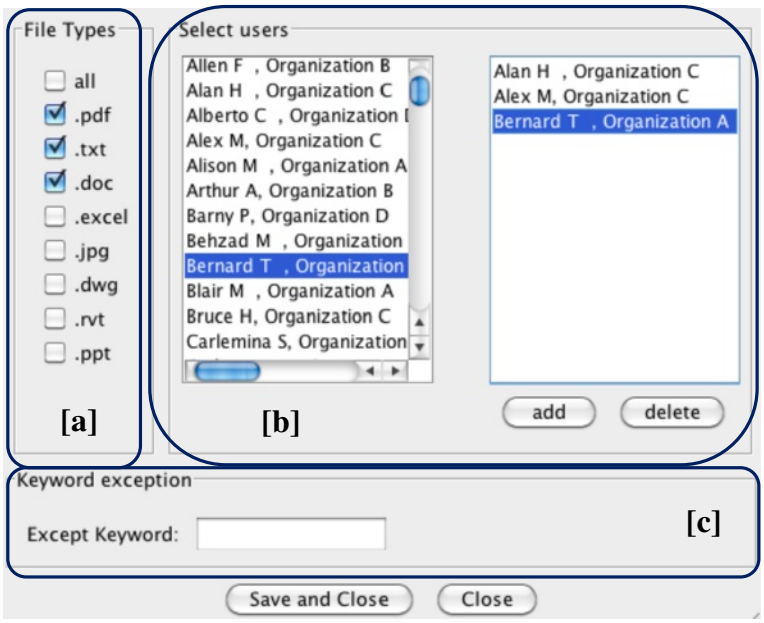

Fig. 3 Advanced filters designed for the construction file archives: (**a**) File types; (**b**) User who created the file; (**c**) Keyword exception

project archive and the searching preferences of users. For example, software developers might be interested in searching software defects for specific software components, release versions, etc.

Access history visualization viewer

Users can view access history information of a selected file item using "View Access Info" tab. A pop-up window with similar visualization representation and interaction to that of *VisArchive's* main screen consists of a timeline visualization (Fig. 4(a)) to visualize summary information about the access history, an access history browser (Fig. 4(b)) to display the details of access history, and a user filter (Fig. 4 (c)) to filter the access history by access user name. The access history visualization viewer uses distinct colors to indicate or distinguish visually the different types of access, so that users can recognize when and how a file was accessed.

Relevance algorithm

The relevance algorithm generates the relevance-ranked search results, which are represented visually on the interactive timelines and information browser by applying visualization representations and supporting visual cues. Figure 5 schematically shows the process. It integrates relevance-ranked search results with a visual representation to enable users to visually search and explore the archives more easily and intuitively. This algorithm could be easily replaced by any other ranking algorithm if different relevance criteria were desired.

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. The 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. 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 one 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, we 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. We assign level 5 to the item if five 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 zero up to the number of search keywords. The relevance-ranked search results are processed with visualization techniques and visually represented to users in the user interface. While the current implementation of *VisArchive* can support archives with thousands of files without system performance issues, for

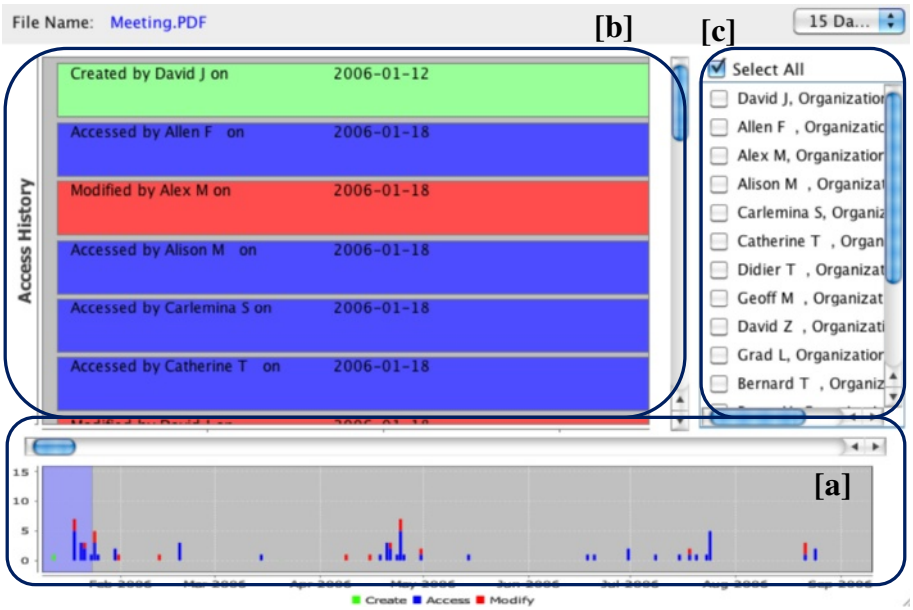

Fig. 4 Visualization showing access history of a selected file: (**a**) Timeline visualization; (**b**) Access history viewer; (**c**) User filter

very large archives, some adaptations to the relevancy algorithm and the user interface may be needed.

Implementation details

VisArchive was implemented as a desktop application using Java and the JFreeChart (2013) visualization toolkit. Most of the charting and visualization used in this prototype were generated by using the JFreeChart API with modifications and customizations. The 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, the extraction of textual information as meta-data for keyword-based searching from the electronic files is needed. For ease and efficiency of generating a dataset to demonstrate the concept used in the interface, this information was extracted and created manually from the existing archives. From the construction project archive, electronic files were indexed by extracting all necessary meta-information regarding each document and integrating this information into the database for demonstration purposes. The meta-data that were 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.

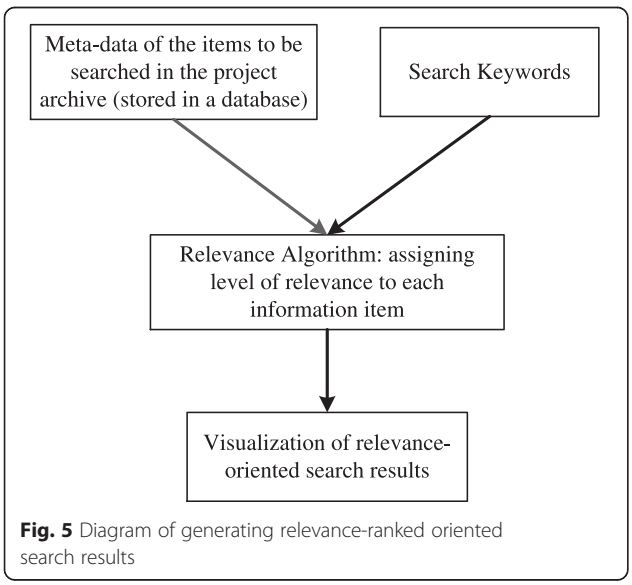

Fig. 5 Diagram of generating relevance-ranked oriented search results

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 the 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.

Evaluation and Discussion

In order to examine the feasibility of using *VisArchive* for searching, browsing and exploring information in

project archives of different domains, two case studies were conducted. The case studies involved (1) a construction project archive, and (2) a software project archive (software defect tracking). Specifically, the design principles implemented in *VisArchive* were examined in relation to the three context-specific design features: (1) the temporal context shows timeline based visualizations; (2) the search-relevance context shows visual indications of search relevance and matched keywords; and (3) the usage context shows visual representations of user activity in shared repositories. The evaluation focused on the feasibility of the prototype to resolve complex use scenarios, rather than simple search using known file names or IDs.

The interface of *VisArchive* was revised and modified based on these case studies, but the core features of the tools described in Section "System design details, reasoning structure and implementation" remained stable. 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. The information browser for the software project case study was modified to display the summary for each software defect. Users can identify a software defect item by viewing its summary.

Case study 1: construction project archives

In this case study, *VisArchive* was used to search construction project archives of an educational building project. The project archive contained more than 800 files that were created during its design development phase by different individuals involved in the project. We created a project archive using 300 files that were selected based on the information available for testing the 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. The information, such as ID, name, path, and description of each file, was archived as searchable meta-data into a database system for archive searching, data processing and visualization by *VisArchive*.

This case study provides access history visualization components for visualizing file access history in the archive. Since prior file access history data was not provided by the construction project archive, a synthetic file access history for a number of files in the database was created for demonstration purposes. The testing focused on searching and exploring the files in the archive that users might have never accessed before, or that users lacked specific information about the files.

Searching files that match all the search keywords

The testing considered the real scenario in which a project manager (PM) wanted to find all "electrical", "mechanical" and "structural" documents that an engineer (named hereafter as "Mike") had worked on. The PM had to share them with another engineer that came onboard after Mike left the company. Traditional search methods present search results as a list of files from top to bottom, with information such as file name, size, last-modified date, etc. Although existing search solutions such as Buzzsaw® enable the most relevant files to appear at the top of the list, the PM would not know clearly which keywords and how many of them were matched. Accordingly, the PM might need to open each file to evaluate how relevant it is to the search keywords. It would also have been difficult for the PM to understand how these files had been produced along the way. This is important because the PM needs to find files that were produced in a certain period in the project's history.

For the scenario described above, *VisArchive* enables the PM to easily identify the files matching all the search keywords ("electrical," "mechanical," "structural," and "Mike") on ten different dates. These files are considered the most relevant to the PM's search and contain *all* the search keywords entered. Thus, *VisArchive* helps the PM to view the most relevant files first — more quickly than otherwise possible when searching manually. The efficiency of finding the most relevant information is very critical particularly in a large-scale construction project archive. The blue arrows in the timelines not only indicate the relevant files, if any, in the archive — matching all the search keywords — but also provide users a visual overview of when these files were created during the specific project stage (Fig. 6). In order to help the PM to retrieve the most relevant file out of the search results, the information about each file in the information browser is very useful as it allows the PM to view and access detailed information about the files (Fig. 7).

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. For example, the PM in the above-mentioned scenario, besides searching for the files that match all the search keywords ("electrical," "mechanical," "structural," and "Mike"), was further interested in exploring other files that matched to one or more of these keywords. For example, the PM needed information about other "electrical" related files which Mike was also involved in. Existing search solutions for construction project archives make it difficult

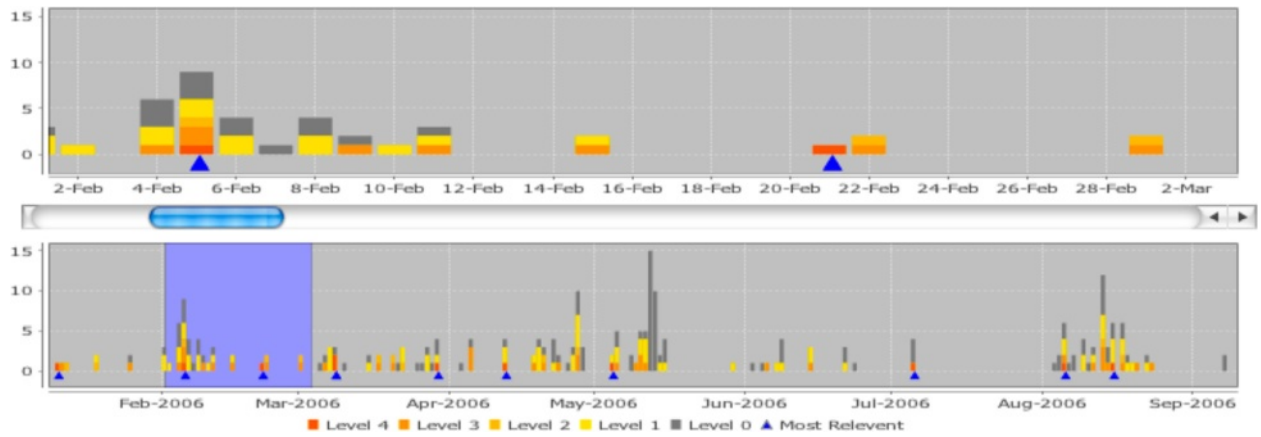

Fig. 6 Timeline visualization while searching keywords: electrical, mechanical, structural, and Mike

for individuals to explore the files by their relevance to the search keywords as they generate a long list of files. Since there is no visualization of the search results, 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 and query for information such as the following: Are there any files that match a certain number of the search keywords? When were these files created in the project timeline? Which month contains more relevant files than the others? The color-scaled visual support both in the timeline and the information browser of *VisArchive*, allows the PM to identify these relevance-ranked files and to identify the level of relevance for each file in the information browser. For example, two files (Fig. 7 (a) and (b)) are shown as less relevant than the most relevant file (Fig. 7 (c)), but they are 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. For example, when the PM wanted to explore electrical documents with which Mike was involved (files containing "electrical", and "Mike"), other files relevant to other search keywords are also shown with the same relevance as shown in Fig. 7, (e.g. files containing "structural" and "Mike"). With existing solutions, users would need to read extra meta-information of each file in order to differentiate between files with the same relevance level.

Besides searching and exploring files in the project archive, the PM wanted to explore other information,

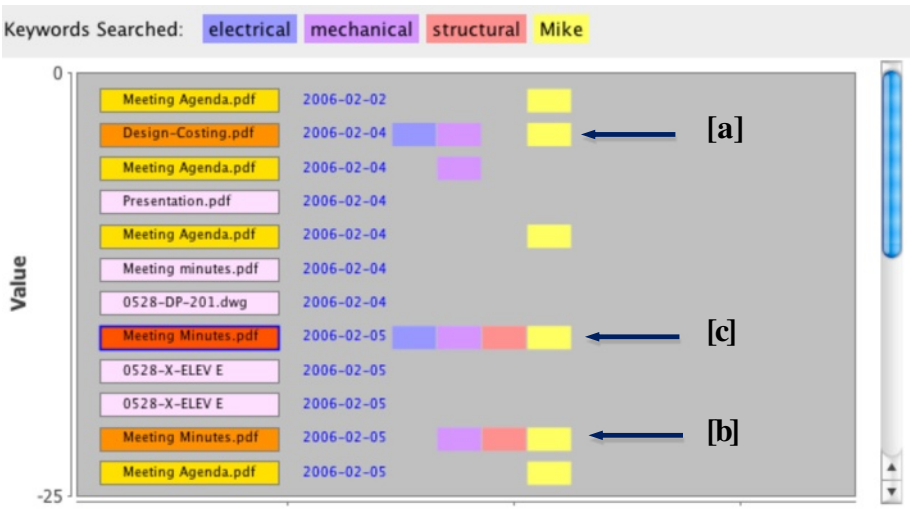

Fig. 7 Information browser displaying the files with color-coded visual supports: (**c**) most relevant file; ((**a**) and (**b**)) second most relevant files

for example, identifying the time periods in which the project archive was more active (i.e. when more files were created). 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. Because the timelines convey information about the various activities and file types created during the project (e.g. documents such as different layout plans may have been created most frequently earlier on in the project), *VisArchive* can help to narrow down the time periods and the intensity of project activities in order to find the most relevant documents.

Exploring file access history

The access history information of files in the constructiion archive is extremely important for design coordination and development. The architect on the project had placed the latest and updated version of the architectural design into a shared construction archive but was not sure whether the consultants had accessed it. 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). In the access history viewer (Fig. 8), 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). The access records can be filtered by individuals who have created, accessed, or/and modified the file.

Case study 2: defects tracking of mozilla thunderbird project

The second case study expands the application of *VisArchive* beyond the construction project archive to explore software defects tracking in the open-source software development domain. Unlike the construction project archives, software defects in this case study were not structured into directories as digital files. Compared to the previous case study, this case study project provides a test environment for searching and visualizing larger amounts of unstructured data.

The Mozilla project was started in 1998 and was intended to develop open-source software projects using the power of thousands of programmers all over the world (Mozilla project 2014). 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 (Bugzilla 2014) 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. Around 1000 defect records from Bugzilla under the Thunderbird project between 2006 and 2007 were used for testing. A modified and simplified version of the *VisArchive* interface was used compared to the construction case study. For each software defect, the defect ID, date, and summary were used as meta-data within *VisArchive*.

Bugzilla allows users to search the defect archives by entering keywords and using advanced filters that are similar to the search mechanism of *VisArchive*. Finding relevant information over thousands of defect records in

Fig. 8 Visualization of access history filtered by selected access users

Bugzilla is a tedious process. Moreover, the search results are represented in a conventional list of defect information (Fig. 9). 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.

The defect summary was described by the defect finder, and contained 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, emphasis in this case study was on finding relevant software defects by searching and exploring through the summary of software defects.

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 responsibility. In addition, time information, such as when the issues were filed, is also useful for developers to prioritize them and fix. Figure 10 shows an example of search results and visualization support that is provided by *VisArchive* for the software defect project. The software developer can take advantage of the timeline visualization of search results to easily identify software defects along with the date that these defects were created. With *VisArchive*, the developer can navigate to the time range containing the earliest and most relevant defects found (Fig. 10(a)) in the timeline and view the defect summary of the most relevant defect (Fig. 10(b)) in the information browser. Regardless of the size of the archive, the blue arrows always provide awareness of the most relevant results and accordingly offer visual cues to the users.

Exploring the defect archive and relevant software defects

If none of the defects matches the user's requirements 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 identify the most relevant defects in the archive and where these defects occur on the archive timelines.

Figure 11 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 (Fig. 11 (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

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

Fig. 9 Conventional list of search results provided by Bugzilla

Fig. 10 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

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 (Figs. 11 (b) and 12(a)), whereas the defects containing "window file" may be disregarded (Fig. 12(b)).

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

Fig. 11 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"

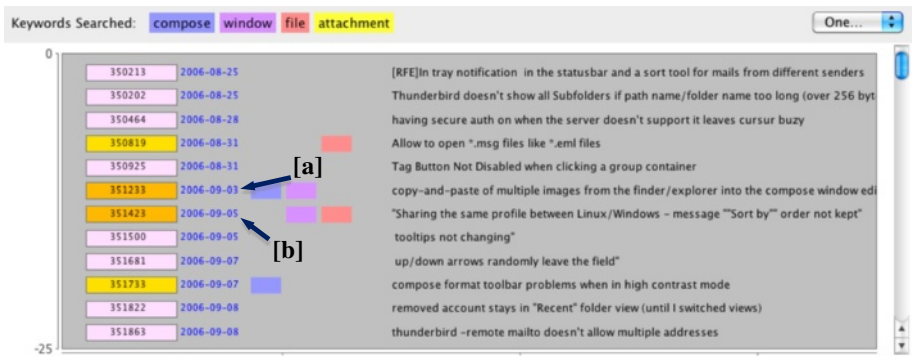

Fig. 12 Sample information displayed in information browser: (**a**) the defect matching "compose window" (**b**) the defect matching "window file"

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, labelled "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 that it has since been resolved. This information could be helpful to the developer to explore whether the resolution to the earlier bug logged in May could help to resolve the similar bug detected in August.

Conclusions

This research extends query-driven interface research by providing three context-specific design features: timeline based visualizations; visual indications of search relevance and matched keywords; and visual representations of user activity in shared repositories. These design features were implemented in an interactive visualization tool called *VisArchive* that integrates multiple commonly used visualization and user interaction techniques to facilitate searching, browsing, and exploring information in historical project archives. *VisArchive* visualizes relevance-ranked search results with color-coded stacked bar charts in project timelines and uses additional supporting visual cues to distinguish search results based on search keywords. Two case studies from the construction and software project domains were used to demonstrate its applicability, usefulness and generality.

Currently, *VisArchive* allows users to view the access history of a single item (e.g. a file or a defect record) in the project archive. It might be useful for users to also explore the access history of multiple items in the archives. Different visual representations might be used to achieve this goal in future research (e.g. multiple timelines could be used to represent the access history of different files in one display, or they could be aggregated into one timeline and use color-scale to distinguish different items). The interactive timeline visualization could also be more flexible to allow for different time frames. The current implementation represents each bar of the timeline in terms of the number of items created in 1 day. However, if the project archive covers a long period of time (e.g. the lower timeline of the software defect archive, visualizing around 1000 records over 2 years), the bar chart will be compacted, and the user will have difficulty clearly seeing the color-coded visualization in the lower timeline. Furthermore, the relevance-ranking algorithm for generating search results is based on keywords, which may limit the quality and reliability of the search results in other domains.

The current prototype used manually extracted meta-data for demonstration purposes. However, 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 embedded to the system in place of the conventional text based search results to provide better visualization and

interaction capabilities. However, in a file or document based archive such as the construction archive, searching can be more challenging. To overcome this, 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 the keywords searchable in the system. The quality of the search results will depend on the searchable meta-data and the user's analytical ability.

While the visual cues for both the historical context and the matching-level 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 whether users can indeed intuitively understand the meaning of visual indicators implemented in *VisArchive*. In addition, user studies are also needed to examine the effectiveness, usability and user experience of *VisArchive* in various domains. We believe that the design principles implemented in *VisArchive* can be applied to different domains as long as the information items in the archive have temporal information, such as creation date, and the relevant meta-data. *VisArchive* simply requires the access information (e.g. creation date, access date, and modification date) and searchable meta-data (e.g. summary, description, keywords, tags) to function. For example, research papers in the IEEE online archives (IEEE 2014) 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. Advanced filters and an information browser could be customized and modified based on users' needs in these different domains.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

In preparing this manuscript, all the listed authors have made significant contribution. KH carried out the gap analysis, literature review, research design, implementation, and evaluation studies. MT and SS-F conceived the research idea, defined the problem and assisted in the design, implementation and evaluation of the research. MN extended the literature review and gap analysis, and helped to refine the concepts and draft the manuscript. All authors have read and approved the final manuscript.

Received: 8 August 2015 Accepted: 22 February 2016
Published online: 22 March 2016

References

Ahlberg, C., & Shneiderman, B. (1994). *Visual information seeking: tight coupling of dynamic query filters with starfield displays* (pp. 313–317). Boston: Pro. of the SIGCHI Conference on Human Factors in Computing Systems.

Ahlberg, C., Williamson, C., & Shneiderman, B. (1992). *Dynamic queries for information exploration: an implementation and evaluation* (pp. 619–626). Monterey: CHI'92 Proc. of the SIGCHI Conference on Human Factors in Computing Systems.

Alonso, O., Gertz, M., & Baeza-Yates, R. (2009). *Proc. CIKM'09, November 2–6, 2009* (pp. 97–106).

Bugzilla. (2014). https://bugzilla.mozilla.org/. [Accessed 6 Mar 2016].

Buja, A., McDonald, J. A., Michalak, J., & Stuetzle, W. (1991). *Interactive data visualization using focusing and linking* (pp. 156–163). San Diego: IEEE Conference On Visualization '91.

Buzzsaw. (2014). http://www.autodesk.com/products/buzzsaw/overview. [Accessed 6 Mar 2016].

Caldas, C. H., Soibelman, L., & Han, J. (2002). Automated classification of construction project documents. *Journal of Computing in Civil Engineering, 16*(4), 234–243.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in information visualization: using vision to think*. San Francisco: Morgan Kaufmann Publishers.

Demian, P., & Balatsoukas, P. (2012). Information retrieval from civil engineering repositories: importance of context and granularity. *Journal of Computing in Civil Engineering, 26*(6), 727–740.

Demian, P., & Fruchter, R. (2006a). An ethnographic study of design knowledge reuse in the architecture, engineering and construction industry. *Research in Engineering Design, 16*, 184–195.

Demian, P., & Fruchter, R. (2006b). Methodology for usability evaluation of corporate memory design reuse systems. *Journal of Computing in Civil Engineering, 20*(6), 377–389.

Dork, M., Carpendale, S., Collins, C., & Williamson, C. (2008). VisGets: coordinated visualizations for web-based information exploration and discovery. *Visualization and Computer Graphics, IEEE Transactions on, 14*(6), 1205–1212.

Dumais, S., Cutrell, E., Cadiz, J. J., Jancke, G., Sarin, R., & Robbins, D. C. (2003). *Proc. SIGIR'03* (pp. 72–79).

Fails, J. A., Karlson, A., Shahamat, L., & Shneiderman, B. (2006). *A visual interface for multivariate temporal data: finding patterns of events across multiple histories* (pp. 167–174). Baltimore: IEEE Symposium on Visual Analytics Science and Technology.

Foo, S., & Hendry, D. (2007). *Desktop search engine visualisation and evaluation* (10th International Conference on Asian Digital Libraries, pp. 372–382). Hanoi: Springer Verlag.

Froehlich, J., & Dourish, P. (2004). *Unifying artifacts and activities in a visual tool for distributed software development teams* (pp. 387–396). Edinburgh: Proc. of the 26th International Conference on Software Engineering.

Gilbert, E., & Karahalios, K. (2007). *CodeSaw: A social visualization of distributed software development* (11th IFIP TC 13 International Conference on Human-Computer Interaction, INTERACT, pp. 303–316). Rio de Janeiro: Springer Verlag.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., de Melo, G., & Weikum, G. (2011). *Proc. WWW 2011* (pp. 229–232).

IEEE. (2014). http://www.ieee.org/index.html. [Accessed 6 Mar 2016].

Indratmo, Vassileva, J. Gutwin, C. (2008). *Exploring blog archives with interactive visualization*. Naples: Proc. of the Working Conference on Advanced Visual Interfaces.

Isenberg, P., & Fisher, D. (2009). Collaborative brushing and linking for Co-located visual analytics of document collections. *Computer Graphics Forum, 28*(3), 1031–1038.

JFreeChart (2013). http://www.jfree.org/jfreechart/. [Accessed 6 Mar 2016].

Jones, R., & Diaz, F. (2007). Temporal profiles of queries. *ACM Trans. Information Systems, 25*(3), 31.

Jones, S., Payne, S. J., Hicks, B. J., & Watts, L. (2013). *Visualization of heterogenous text data in collaborative engineering projects*. Atlanta: The 3rd IEEE Workshop on Interactive Visual Text Analytics.

Kim, S.-A., Choe, Y., Jang, M., & Seol, W. (2011). *Design process visualization system intergrating BIM data and performance-oriented design information*. Seoul: 28th International Symposium on Automation and Robotics in Construction.

Knowledge and Information Management (KIM). (2006). KIM grand challenge project. http://www.ukoln.ac.uk/projects/grand-challenge/. [Accessed 6 Mar 2016].

Krishnan, A., & Jones, S. (2005). TimeSpace: activity-based temporal visualisation of personal information spaces. *Personal and Ubiquitous Computing, 9*(1), 46–65.

Kumar, V., Furuta, R., & Allen, R. B. (1998). *Metadata visualization for digital libraries: interactive timeline editing and review* (pp. 126–133). Pittsburgh: Proceedings of the 1998 3rd ACM Conference on Digital Libraries.

Kwon, B. C., Javed, W., Ghani, S., Elmqvist, N., & Ji, S. Y. (2012). Evaluating the role of time in investigative analysis of document collections. *IEEE Transactions on Visualization and Computer Graphics, 18*(11), 1992–2004.

Mozilla Project. (2014). https://bugzilla.mozilla.org. [Accessed 6 Mar 2016].

Ogawa, M., & Ma, K.-L. (2008). *StarGate: A unified, interactive visualization of software projects* (pp. 191–198). Kyoto: IEEE Pacific Visualization Symposium.

Padia, K., Alnoamany, Y., & Weigle, M. C. (2012). *Visualizing digital collections at archive-it* (pp. 15–18). Washington: 12th ACM/IEEE-CS Joint Conference on Digital Libraries.

Pirolli, P., Card, S. K., & Van Der Wege, M. (2001). *Visual information foraging in a focus + context visualization* (pp. 506–513). Seattle: Proc. of the SIGCHI Conference on Human Factors in Computing Systems.

Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., & Shneiderman, B. (1998). *LifeLines: using visualization to enhance navigation and analysis of patient records* (Proceedings of the AMIA Symposium, pp. 76–80).

Rezgui, Y. (2001). Review of information and the state of the art of knowledge management practices in the construction industry. *Knowledge Engineering Review, 16*(3), 241–254.

Russell, A. D., Chiu, C.-Y., & Korde, T. (2009). Visual representation of construction management data. *Automation in Construction, 18*(8), 1045–1062.

Steed, C. A., Symons, C. T., DeNap, F. A., & Potok, T. E. (2012). *Guided text analysis using adaptive visual analytics*. Burlingame: Proc. SPIE 8294, Visualization and Data Analysis.

Storey, M. D., Čubranić, D., & German, D. M. (2005). *On the use of visualization to support awareness of human activities in software development: A survey and a framework* (Proc. of the 2005 ACM Symposium on Software Visualization, pp. 193–202).

Strotgen, J., & Gertz, M. (2012). *Event-centric search and exploration in document collections* (pp. 223–232). Washington: Proc. of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries.

Tory, M., Staub-French, S., Po, B. A., & Wu, F. (2008). Physical and digital artifact-mediated coordination in building design. *Computer Supported Cooperative Work, 17*(4), 311–351.

Veerasamy, A., & Heikes, R. (1997). *Effectiveness of a graphical display of retrieval results* (pp. 236–245). Philadelphia: Proc. of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Viegas, F. B., Golder, S., & Donath, J. (2006). *Visualizing email content: portraying relationships from conversational histories* (pp. 979–988). Montreal: Proc. of the SIGCHI Conference on Human Factors in Computing Systems.

Walk, S., Pöschko, J., Strohmaier, M., Andrews, K., Tudorache, T., Noy, N. F., et al. (2013). PragmatiX: an interactive tool for visualizing the creation process behind collaboratively engineered ontologies. *International Journal on Semantic Web and Information Systems, 9*(1), 45–78.

Wu, F., & Tory, M. (2009). *PhotoScope: visualizing spatiotemporal coverage of photos for construction management* (pp. 1103–1112). Boston: 27th International Conference Extended Abstracts on Human Factors in Computing Systems.