Ontology-assisted Provenance Visualization for Supporting Enterprise Search of Engineering and Business Files

Saiful Khan[a,*], Urszula Kanturska[c], Tom Waters[c], James Eaton[c], René Bañares-Alcántara[a], Min Chen[b]

[a] *Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK*
[b] *Oxford e-Research Centre, University of Oxford, 7 Keble Road, Oxford, OX1 3QG, UK*
[c] *Laing O'Rourke, Admirals Park, Crossways, Dartford, DA2 6SN, UK*

Abstract

In many large engineering enterprises, searching for files is a high-volume routine activity. *Visualization-assisted search* facilities can significantly reduce the cost of such activities. In this paper, we introduce the concept of Search Provenance Graph (SPG), and present a technique for mapping out the search results and externalizing the provenance of a search process. This enables users to be aware of collaborative search activities within a project, and to be able to reason about potential missing files (i.e., false negatives) more effectively. We describe multiple ontologies that enable the computation of SPGs while supporting an enterprise search engine. We demonstrate the novelty and application of this technique through an industrial case study, where a large engineering enterprise needs to make a long-term technological plan for large-scale document search, and has found the visualization-assisted approach to be more cost-effective than alternative approaches being studied.

Keywords:
Enterprise search, engineering document, knowledge management, information visualization, provenance visualization

1. Introduction

With a growing amount of data in enterprises, managing and searching for information is becoming a challenge. In Information Retrieval (IR), the concept of *Enterprise Search* was introduced to encourage real-world application of IR, where an enterprise develops an integrated data infrastructure, and enables its stakeholders to have an effective search interface for information retrieval. The scope of enterprise search has been defined by a number of authors, e.g., Stenmark [1], Abrol *et al.* [2], and Hawking [3]. Many have argued the importance of enterprise search from the perspective of "the high cost of not finding information" [4]. This work focuses on using visualization to reduce the cost in the process of finding information.

It is estimated that the enterprise search industry is growing at 25% per year [5]. A tutorial in ACM SIGIR, 2010 [6] highlighted some of the key challenges in the area of enterprise search. Although these challenges are similar to Web search, many cannot be addressed by existing Web search solutions [6]. For example, the structure of enterprise or desktop data is different from that of Web data. The popular *ranking* technique (e.g., PageRank) is not effective in enterprises search [3]. Although the relevance feedback information can be used to learn and improve the rank of the search results in the enterprise search [7], such techniques often result in an extensive amount of search results but do not guarantee adequate recall and precision. Therefore, there is a need for effective designs of visualization and interaction [8]. In enterprise search, discovering information searched and found by experts and colleagues is often part of a search strategy [9, 6]. Systematic support for such collaborative search effort is thus highly desirable.

File search is an elementary task in enterprise search. In a construction company, for example, a large number of files are captured and stored in each project. These files may be of types of reports, CAD files, presentations, scan copies, spreadsheets, art works, images, audios, videos and so on. It has been reported that stakeholders (e.g., engineers, managers, administrators) spend around 30% of time in file search [10, 11, 12]. However, searching through a large data infrastructure is a time consuming operation, as (i) search queries often return a long list of files, (ii) the list usually includes false positives, and (iii) the queries often miss some target files (false negatives).

Previous research have been focused on different mechanisms to improve the search time by enhancing the retrieval capability of enterprise search engines for engineering document [11, 12], multi-topic documents [13], patents [14], mechatronic data [15], BIM document [16, 17, 18] and so on. However, existing search engines typically do not provide information about historical search activities. Even when it was done occasionally, the information would be provided in a log form, and it would be time-consuming to read and comprehend such data. In this

*Corresponding author, Phone: +44-7769568866
Email addresses: saiful.khan@eng.ox.ac.uk (Saiful Khan), UKanturska@laingorourke.com (Urszula Kanturska), TWaters@laingorourke.com (Tom Waters), JEaton@laingorourke.com (James Eaton), rene.banares@eng.ox.ac.uk (René Bañares-Alcántara), min.chen@oerc.ox.ac.uk (Min Chen)

work, we report a novel approach for using visualization to support collaborative search activities, utilizing both machine and human capabilities.

File search is an explorative process, during which a user progressively refines search criteria by observing and learning from successes and errors in the earlier search results. Devising effective search criteria is a complex cognitive process, which depends on many factors, such as individual users' knowledge about the files to be searched, their familiarity of the organization of the file repository, the critical nature of the search tasks, the likelihood of missing files or multiple copies of the same files, and so on. Visualization-assisted search tools have been found to be effective in supporting search activities, e.g., assisting in iteratively reformulating queries and visualizing search results [19].

In an industrial search operation for engineering and business documents, search tasks are often performed by a team over a period. In order to enable such tasks to be performed efficiently and effectively, it is necessary for team members to access and share the *provenance information* [20] about the historical and ongoing activities related to a task. For complex provenance records, visualization can provide an enabling tool for users to observe provenance information at a glance [21], improving the cost-effectiveness in in collaborative search tasks, especially in an industrial setting.

This work was conducted in the context of an industrial application, where intensive file search is part of an industrial process and is to be performed by a team on a daily basis. The background requirements were defined several years ago for the construction industry. By 2016, for a large number of building projects, a construction company must pass on a collection of documents to the client when a building is completed [22], enabling search and reuse of the digital assets. As the total number of files related to a building project could easily be in tens or hundreds of thousands, and many are distributed among different teams and stakeholders in the company, the need for file search in a large distributed file system becomes inevitable.

As part of a feasibility study led by a large construction enterprise, we developed a prototype system that provides visualization-assisted search capabilities. The objective is to enable the enterprise to assess the merits of visualization in comparison with two other approaches, database without visualization and search engine without visualization. This paper presents the visualization-assisted search system, and reports our experience in delivering a technical solution for a challenging industrial problem.

In this work, we addressed several requirements that cannot currently be met by existing enterprise search systems (e.g., Lucene [23], Solr [24], and Indri [25]). These requirements include:

(a) It is desirable to capture, retrieve, and visualize the provenance records of project-based search processes that feature continuing and collaborative search activities lasting for days or weeks. To accomplish this:

- We have introduced a graph structure called *Search Provenance Graph (SPG)* for recording the search provenance in individual projects and for enabling provenance visualization.
- We have developed a combined glyph-graph visual representations for visualizing SPGs in a focus+context manner.

(b) It is necessary to couple a visualization interface with a search engine, and to have a common semantic framework for the knowledge base in a search engine and analytical processing in visualization. For example, to measure the similarity between two search records, it is essential to ensure that the same measurement is used by the search engine and provenance visualization. Because it is not feasible to access the knowledge base within a commercial search engine, we had to develop a research search engine. For this purpose,

- We have developed a prototype *enterprise search engine* with *visualization-assisted search capabilities* for supporting collaborative search activities, query formulation and reformulation, and identification of false positives and false negatives.
- We have developed *multiple ontologies* as a knowledge base for supporting the search activities as well as for computing semantic similarity in constructing and updating SPGs for provenance visualization. Ontologies stores concepts and instances, along with their relations and properties as a knowledge-base [26].

(c) It is beneficial to evaluate the proposed technical solution in the context of a real world application. To achieve this:

- We have worked closely with a management team in the industrial partner to compare three approaches for supporting enterprise search operations, and assisted the industrial partner in making a strategic plan for meeting the new requirements to be implemented in 2016. Our visualization-assisted approach compared favorably with alternative approaches under investigation.

The rest of the paper is organized as follows. In Section 2 we present an overview of the related literature. In Section 3 we describe the main motivation behind this work, and a real world industrial scenario. In Section 4 we provide a formal definition of the proposed SPG. In Section 5 we briefly describe an enterprise search engine developed to enable the implementation of SPG and the visualization facilities. In Section 6 we propose an algorithm to compute the similarity between queries in SPG. In Section 7 we present glyph-based visualization and provenance visualization techniques developed for this system. In Section 8 we report an evaluation of the visualization facilities by the domain experts. In Section 9 we offer our concluding remarks.

2. Related Work

Our work builds mainly on two areas of research *(a) Information Retrieval (IR):* search strategies, query reformulation, and query similarity measurement, and *(b) Visualization:* visualization-assisted search, provenance visualization, ontology-assisted visualization, and glyph-based visualization.

2.1. Search-related Visualization

Many research papers have been published by the visualization and IR communities on navigation and exploration of the Web information space.

VisGets [19] was developed to visualize different Web search attributes, to assist in formulating search queries and to visualize search results using interactive geographic maps and tag clouds. Applications of VisGets to a variety of Web data collections were demonstrated in [27]. An empirical study on interactive and visual exploration of the Web using VisGets was reported in [28]. Fluid Views [29] allows users to view Web search results geographically and temporally in dual layers. PivotPaths [30] allows users to explore datasets with a variety of relation types across various dimensions. A Voronoi treemap to organize search results was proposed in [31]. ProjSnippet [32] visualizes web search results in a global view and provides a clustering technique for identifying similar content. The similarity measurement used in this work is based on the vector space model. Footprints [33] was developed for analysts to help them search documents and articles.

2.2. Provenance Data Visualization

The visualization technique commonly used for provenance data is the node-link diagram [20]. For example, such diagrams are used for visualization workflows in VisTrails [34, 21, 35], scientific workflows in [36], and biological experiment workflows in [37, 38]. They are also used in provenance management tools such as Haystack [39], Probe-It [40], and Orbiter [41]. These node-link diagrams feature a variety of visual encoding methods, such as including directed graphs [39, 40], collapsible summary nodes [41], and glyph representations [37, 38]. Dunne *et al.* proposed visualization of analytic provenance in [42]. Borkin *et al.* [43] proposed the InProv tool which employ a radial layout with hierarchical encoding of the time dimension. Walker *et al.* proposed a XML-based provenance model to support data exploration and to visualize provenance in the context of Human Terrain Analysis in [44].

2.3. Search Provenance or Log Visualization.

Web search activities generate query logs [45], usability logs [46], and web clickstreams [47]. The SDSS Log Viewer [45] was developed for identifying users' data-seeking behaviors from SQL query logs. Clickstreams [48, 49] supported the analysis of Web traffic and users Web path navigation behavior [50]. Behavior Graphs [48] used state diagrams to visualize search structures on the Web for discovering usage patterns. WebQuilt [51] and Webviz [52] used trees and node-link diagrams to visualize click-streams. Lee *et al.* [49] used parallel coordinates and star fields to visualize user paths and product performance. Wei *et al.* [47] introduced an interactive clustering method to reveal user behavior patterns from Web click-streams data. TrailExplorer2 [53] used stacked bars and pie charts to discover valuable information from large-scale Web click-streams. Lam *et al.* [54] provided users with multiple coordinated views for analyzing Web search behaviors. Shi *et al.* [55] focused on analyzing the dynamics of user loyalty and defection using Web log entries. Stuff I've Seen [56] was developed to provide a unified access to provenance of what a user has seen on their desktop.

2.4. Ontology-assisted Visualization

There is a substantial amount of literature on visualizing ontologies and navigating their structures. Many of them can be found in the survey paper [57]. The need to build an ontology of visualization to capture the concepts and characteristics of visualization was discussed in [58]. VisIOn [59] was developed for categorizing and storing information about software visualization systems. Gilson *et al.* [60] proposed to use multiple ontologies to enable automated generation of visualization. Carpendale *et al.* [61] discussed several research directions in ontology-assisted visualization. Ontology-assisted visualization has also been developed in applications such as biology [62, 63, 64], Big Data [65], social networks [66], geography [67] and so on.

3. Case Study: Enterprise File Search

This research was motivated by a significant industrial challenge in the construction industry. As part of Building Information Modeling (BIM) [22, 68], the industry is expected to meet a set of new requirements for providing clients with the relevant data about a building and the building process at the end of each building project. While the industry is formulating a detailed specification about what documents and in what formats, it has been developing strategies for meeting the requirements in a cost-effective manner.

This collaborative work with a large construction enterprise started some three years ago. We have studied the documents about the requirements, and watched videos featuring discussions about these requirements among senior figures in the industry. We considered two main strategies for addressing these requirements, namely (a) requiring each building project to maintain a high quality database of all documents, from which a collection of the required files can easily be found at the end of the project; (b) requiring all files to be accessible by a search engine, which will be used by a team of administrators to compile a collection of the required files when they are needed.

As part of the overall evaluation of these two strategies, a university research project was formulated to study the feasibility of strategy (b). In particular, the focuses have been placed on two research questions: what technical features of search engine are required, and how visualization can be used to support file search?

It is not difficult to recognize that the construction industry has a large number of stakeholders (e.g., clients, architects, suppliers, local governments, transport authorities) and many teams within each construction enterprise. Many building projects last for years. The files accumulated during each

project may be distributed in different parts of a large file system. Each file may have different versions. Each team or stakeholder may have different name convention. Some projects may share a certain number of files. Errors in file naming and placement are also expected. All these suggested that strategy (b) might not be a winning strategy, though strategy (a) also has some shortcomings such as the commitments of data archivists for each project, and the risk of a single-point of failure within each project.

3.1. Real World Scenarios

To help us appreciate the scale of file search activities, our industrial partner provided us with statistics about several example projects.

- *Education Building Project* — In this £54-million project, a total of 100,047 files were collected on the server. 8,677 files (8.7%) were recorded in a database. 4 files were issued to the client based on database search.
- *Large Infrastructure Project* — In this £200-million project, a total of 461,711 files were collected on the server. 15,538 files (3.4%) were recorded in a database. 2,626 files were issued to the client based on database search.
- *Office Building Project* — In this £250-million project, a total of 591,906 files were collected on the server. Out of them only 45,000 files (7.6%) were recorded in a database. 6,566 files were issued to the client based on database search.

The above real-world scenarios highlighted several challenging requirements. Firstly, although the database has likely captured most important files, more than 90% files are not searchable through the database. The poor coverage by the database was largely due to the effort required to enter a file record for each file into the database.

Secondly, from 2016, the files to be issued to a client will likely be in thousands if not more.

Thirdly, when a search engine was used instead of a database, we do not require any database input and administration effort, though the search would take longer time.

For such a high number of files to be found, the cost of search activities is not trivial. Using a conventional enterprise search engine without provenance capture, retrieval and visualization, one could imagine a variety of difficulties. For example, when a search engine returns hundreds or thousands, it is difficult for a user to decide how many files to investigate, judge what that may have been seen before, and guess what may not be among the returned results. For thousands of files to be found, these activities will continue for a period and may require collaborative teamwork. It is difficult for a user to pick up search activities after a weekend or from another colleague.

3.2. Cost Analysis

As part of the requirement analysis, we conducted cost analysis on three possible approaches to address the routine operational need for searching a large number of engineering and business files. These three approaches are (a) *Database of Files*, (b) *Enterprise Search Engine*, and (c) *Visualization-assisted Enterprise Search Engine*. Approach (b) assumes that files will be recorded in a database, with most semantic metadata (e.g., keywords, authors) will be entered by an administrator throughout a project life-cycle. Approaches (b) and (c) assume that all files are stored and accessible to a search engine, while there is no need for regular database input by an administrator. Most of the effort of search will be concentrated at the end of each engineering project in order to return an appropriate set of files to the client. Approach (c) will be supported by visualization capacity as detailed in this paper and in [69], while (b) will be supported by a textual user interface of conventional search engines.

With the help of cost-analysis experts, we estimate the cost of filing and search for all approaches and database input for (a). The numerical values of the costs reflect typical business costs and are estimated by taking averages over a period of time, e.g., the average of how many files would be generated, how many files would be stored in the distributed file system, how many files would be entered in the database, how many files would be searched and delivered to the customer, and so on. The costs include typical overheads such as training meetings, coffee-breaks, and vacations. They are not measurements of a single task or a button pressing exercise where a task is performed by pressing a button to indicate users' response.

Using the Office Building Project as an example, if entering a file record into a database takes 40 minutes on average (including some reading, figuring out metadata, record maintenance and overhead cost), entering 45,000 files would have taken 30,000 person-hours or 3,750 person-days (i.e., 8 person-hours per person-day). If all 591,906 files had to be recorded in the database, it would take nearly 49,326 person-days.

On the other hand, if one takes 2 minutes on average to find a file in the database to be issued to a client, 6,566 files would take about 219 person-hours or about 27 person-days. Meanwhile, searching without a database would take longer time. If one takes 15 minutes on average to find a file using a search engine, 6,566 files would take about 1,642 person-hours or about 205 person-days. It is thus highly desirable to use visualization-assisted search to reduce the costs involved.

We consider that appropriate visualization can help users to identify false positives and false negatives, provide users with external memorization of search history and enable them to refine their search queries more effectively, and allow a team of users to share information about search activities already performed and to coordinate their activities in an informed manner. Our estimated benefit of visualization can be translated to about 50% reduction of search time. If one takes 8 minutes on average to find a file using a visualization-assisted search engine, 6,566 files would take about 875 person-hours or about 109 person-days.

When we combined all three types of costs (for filing, database, and search), approach (c) was shown to be the most cost-effective solution, which motivated the research described in this paper.

3.3. False Positives and Negatives in Search

One major challenge of strategy (b) is to deal with *false positive* and *false negative* in a search process. Given a set of search criteria, a search engine executes a query, returning a set of results. Among these results, there are:

- **True Positives** — The files have met the search criteria, and are also among those required by the user.
- **False Positives** — The files have met the search criteria, but are not among those required by the user.

Meanwhile, among the vast number of files not returned by the search engine, there are:

- **True Negatives** – The files have not met the search criteria, and are not among those required by the user.
- **False Negative** — The files have not met the search criteria, but are actually among those required by the user.

In the context of this research and its application, it is important for the users to be able to identify and pick out true positives quickly among false positives. It is equally, if not more important, for users to recognize the existence of false negatives [69], and be able to hypothesize where these may be located in the file system and what search criteria may reveal them.

At the early stage of the project, the industrial partner and the university research team had established the following propositions for strategy (b):

- There will be a centralized team of data administrators, who would be responsible for compiling a collection of all required files towards the end of each building project.
- The number of required files was initially estimated at high tens to a few hundreds on average, but later revised to a few thousands. The process should ideally take about 5-10 workings days per project and may involve several administrators.
- The task can be performed collaboratively by several administrators, so sharing of information among them is necessary.
- The administrators will have undergone a training process, and are expected to be familiar with the organization of the enterprise's large file infrastructure, and the search system, including its visual interface.
- The administrators are not expected to know each individual project as well as those reside in the project, and may not know, for instance, the exact file names, dates and locations. Nevertheless, they will have the high-level information about the project, such as the project period, and the major stakeholders and teams involved.

Viewing a textual list of search results has a major shortcoming. It is difficult for a viewer to build a mental model about where these files are, and how they related to each other. For such an observational task [70], visualization can help a viewer in obtaining an overview quickly, and can provide a viewer with external memorization compensating for the rather limited working memory of a human. By reducing the time and cognitive load required for building a mental model and memorization, a viewer can devote more time and cognitive load to thinking (e.g., reasoning false positives and hypothesizing false negatives) [71].

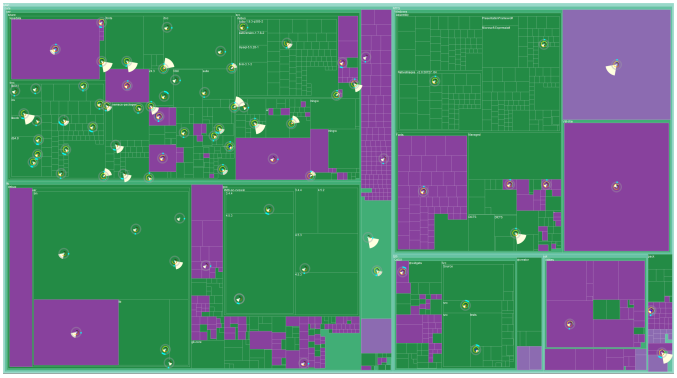


Figure 1: A treemap shows several levels of a distributed file system as the search space, and glyphs represent results of a search with encoding of most important file attributes (e.g., type, date, and size) [69].

At the earlier stage of this feasibility study, we developed a glyph-based visual interface for visualizing search results in conjunction with a treemap representation of the file system [69]. Figure 1 shows a screenshot of such a treemap visualization of research results. This enables administrators to identify false positives based on spatial awareness as well as the glyph representations of the main attributes of individual returned by the search. In addition, the visual representation can also stimulate hypotheses of false negatives. An evaluation in conjunction with this early development showed that static glyphs and animated dots are preferred visual representations of individual files in the search results.

Following this early development, we focus on developing visualization capability that allows users to explore search criteria (i.e., their hypotheses), and to reformulate search criteria progressively.

It is necessary to note that the goal of a user's task is not to formulate the best set of search criteria, but to work efficiently in collaboration to find all required files. Once all the required files have been found, the user will likely stop the process even the best set of search criteria has not been found. In many cases, the best strategy may be to find a complete collection of the required files using a small number of queries even though the search criteria for each query may not be optimal. Meanwhile users' experience of previous projects will play a significant role in formulating search criteria and an effective search strategy. When a search process takes a few days and/or involves more than one administrator, visualization support for search provenance is essential.

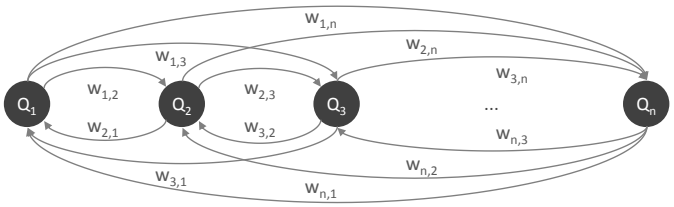


Figure 2: In a traditional query-flow graph, each node represents a query and the weight $w_{i,j}$ of the directed edge from Q_i to Q_j represents the likelihood that the two queries are part of the same search session.

4. Search Provenance Graph (SPG)

We can use a set of inter-connected data records to store the provenance of search activities related to a search task (e.g., finding all required engineering and business documents to be passed onto a client of an engineering project). The connections between records can be made based on search activities (e.g., consecutive searches), reformation of search criteria, or similarity of search criteria (determined algorithmically). These data records can be shared and visualized by members of a search team to facilitate collaboration, such as optimizing search criteria, identifying search space that has been searched and not yet to, and avoiding repeated search activities.

Such provenance information is usually stored in a graph-based data structure. For example, five different types of query graphs were defined by Baeza-Yates [72] in the context of web search: (a) *Word Graph* queries contain the same word(s), (b) *Session Graph* queries belong to a same session, (c) *Cover Graph* user clicked on the same results, (d) *Link Graph* there is a link between the two clicked urls, and (e) *Term Graph* there are k common terms in the contents of the urls.

Boldi *et al.* [73] proposed the *query flow graph* to store statistical and relational information about users' querying behaviors, e.g., interests, preferences, and feedbacks to search results. It is constructed by mining query logs, and it is used for query log analysis, user profiling and personalization, and query recommendation. Figure 2 shows a query flow graph, where each node represents a query and each directed edge from query Q_i to query Q_j indicates the likelihood that the two queries are part of the same search session. The likelihood is given by the weight $w_{i,j}$ associated with each edge.

The above mentioned graph-based concepts cannot be directly used for representing provenance of search activities in the context of this work. Firstly, the provenance in our application is project-based, as described in Section 3. It is important for each administrator or a small team of administrators to be aware of the provenance of search activities related to a specific project, while the search activities within a project do not benefit much from the aggregated statistics about many other projects. Secondly, the provenance data need to provide details such as who initiated what queries and when, who is reformulating whose previous queries, etc. We thereby introduced the notion of *Search Provenance Graph (SPG)*.

In a SPG, each query is represented by a node that is associated with attributes such as search query, time-stamp, search results, and selected results. Each edge indicates a connection

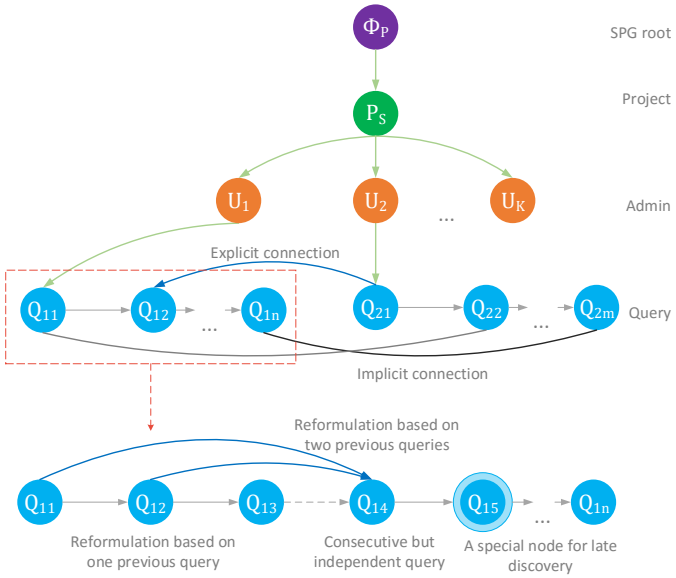


Figure 3: In the overall SPG, there are different types of nodes and edges. The nodes are hierarchically organized.

between two queries. There are two types of connections. An *explicit connection* is typically activated within a search session when a user reformulates a query based on his/her own previous query(s). It can also be activated in a subsequent search session when a user selects another user's search query using the provenance interface as the basis for query reformulation. Hence an explicit connection is inserted by the user interface for search and provenance visualization. An *implicit connection* is inserted into the SPG by a housekeeping program that periodically computes the similarity between nodes in an SPG. Such implicit connections allow users in collaboration to observe similar search criteria that have been performed by different members in a team.

Explicit connections indicate that the administrators formulated queries based on each other, i.e., collaborated well with each other. On the other hand, the degree of implicit connections indicate repeated searches, thus suggesting that collaboration can be improved.

The overall SPG is maintained by the enterprise search system, and it is partitioned into disjoint subgraphs, each of which is associated with a project. Within a project subgraph, there are a list of users (i.e., file administrators), and their search activities are individually recorded.

As illustrated in Figure 3, we can thus define the notion of the SPG as follows. The overall SPG is graph $\Phi_P = \{\mathbb{P}, \mathbb{U}, \mathbb{Q}, \mathbb{E}_{pu}, \mathbb{E}_{uq}, \mathbb{E}_e, \mathbb{W}_i\}$. $\mathbb{P}$ is a set of nodes representing different projects. $\mathbb{U}$ is a set of nodes representing different users within each project. If the same administrator is involved in multiple projects, he/she has a distinct node for each project. $\mathbb{Q}$ is a set of nodes representing the queries performed by users. $\mathbb{E}_{pu}$ is a set of directed edges connecting from each project to its users. $\mathbb{E}_{uq}$ is a set of directed edges connecting from each user to its first query in the project. $\mathbb{E}_e$ is a set of directed edges representing explicit connections. $\mathbb{W}_i$ is a set of undi-

rected weighted edges representing implicit connections. The weight of an edge defines the similarity between two nodes. It is in the range of $[\omega_{min}, 1]$ where $0 < \omega_{min} < 1$ is the minimal similarity value for an implicit connection. Any pair of queries with a similarity value below ω_{min} do not have an edge between them. In this project, we set $\omega_{min} = 0.4$, which can be reconfigured.

The connections from the root to projects are implicitly defined since all projects are connected to the root. There are two types of explicit connections in set $\mathbb{E}_e$. A solid directed edge connecting two nodes in $\mathbb{Q}$ implies that the destination query is a reformulation of the source query. A node with multiple incoming solid edges results from a reformulation based on more than one previous query. A dotted directed edge indicates that the succeeding query is not based on a preceding node though they are temporally consecutive. An undirected translucent edge is an implicit connection. In computer-generated visualization, the similarity is mapped to the opacity of the edge.

There is also a special type of nodes in $\mathbb{Q}$. It is illustrated in dark shade in Figure 3, and represents a set of late discoveries, which are files found by the *Crawler*, rather than the *Query Processing* module.

The overall SPG is dynamically updated during the ongoing search sessions as well as by the background utilities such as the *Crawler* and the housekeeping program for evaluating the similarity between nodes. In order to support such dynamics, many parts of the enterprise search engine have a role to play, which are described in the following section.

5. System Architecture

The commonly used enterprise search engines (e.g., Lucene, Solr, Elastic-Search, Xapian and so on) do not provide search space information to visualize the search results along with the search space to identify the false negatives and false positives [69]. Such general purpose search engines do not provide the search provenance information to study the visualization facilities proposed in this paper. As part of the feasibility study, we designed and prototyped an *enterprise search engine*, enabling the experimentation with visualization facilities, which could not be done with a blackbox search system. The detailed discussion about this research search engine is beyond the scope of this paper. Because the visualization capabilities are closely integrated with several components of the system, we hereby give a brief overview of the system architecture as shown in Figure 4.

The **Vis-assisted Search Interface** a *Search Query Form*, a *Tabular View*, a treemap-based *Focus+Context View*, and a newly-added *Provenance View*. User enters the search criteria in the *Search Query Form*. The search criteria is composed of one or a few keywords and/or some constraints of file attributes, such as type, size and date. The search results are first displayed in a traditional *Tabular View*. A treemap-based *Focus+Context* visualization visualizes the *Search Space* (stored in *Index*) and search results that supports quick *Visual Search* and identification of *False Negatives* which was already reported in [69]. A newly added *Provenance Visualization* facility, the main focus of the paper, supports identification of *False Positives*, *Collaboration*, and *Query Formulation/Reformulation*.

The system consists of a collection of functional modules, which reside between the search interface and the distributed file system.

Query Processor parses the query entered at the *Search Interface*. A set of extended search criteria, e.g., closely-related words, similar file types, etc. is then derived using the *Ontology* module to widen the search space. Thereafter, it searches in the *Index* with the expanded criteria. This also activate the *Crawler* to find files that meet the criteria but are not in the *Index*. Finally, it ranks the retrieved files before returning the results to the *Search Interface*.

Crawler Agents scan through the file system autonomously at a predefined scheduled interval and fetch file information and search space information. It also discover files missed by the *Index*, and place in the "late discoveries" folder to be accessed by the *Provenance Visualization* facility and update the *Index*, and the *SPG Database*.

Index collect files and folders potentially related to future search queries (similar to anticipatory fetch). This supports the *Query Processor* to discover most search results, in real-time. An indexer collects, scans, and parses the meta-data of the files fetched by the *Crawler*. The entire *Search Space* is also cached to visualize the context using a treemap in *Focus+Context* visualization.

SPG Database stores the information required by the SPG: queries performed by the users, the search results received, and the relevant results selected by the user. It provides provenance data to the *Provenance Visualization* facility.

Ontologies provide the enterprise search engine with a knowledge-base. The search engine supports multiple field-based ranking and are usually improved automatically through ontology learning.

A *Term Ontology* stores basic terms in English (including abbreviations) and the weighted relationships between them. It is modeled based on the WordNet [74], with additional weights assigned to all edges. The original WordNet is an unweighted digraph, where each term is connected to set of synonymous terms called a synset. In file search, the words are not necessarily related to each other in exactly the same way as in spoken and written English. We thus introduce a weight to each directed edge, defining the level of "closeness" between two terms. Here the concept of closeness not only encompasses the basic notion of synonyms, but also captures the sense of co-occurrence in searches. For example, "building" and "construction" represent the former, while "work" and "plan" represent the latter.

A *Type Ontology* stores relationships between file types and categories of terms in English. A *Size Ontology* stores relationships between file sizes and categories of terms in English. A *Date-time Ontology* stores relationships between file ages and categories of terms in English.

Nevertheless, the system is deigned to integrate more ontologies if needed, e.g., domain ontology for domain specific semantic support, personalized ontology for personalized information retrieval and so on. At the beginning the weights in the ontologies were set to 0.5. Thereafter, these ontologies has

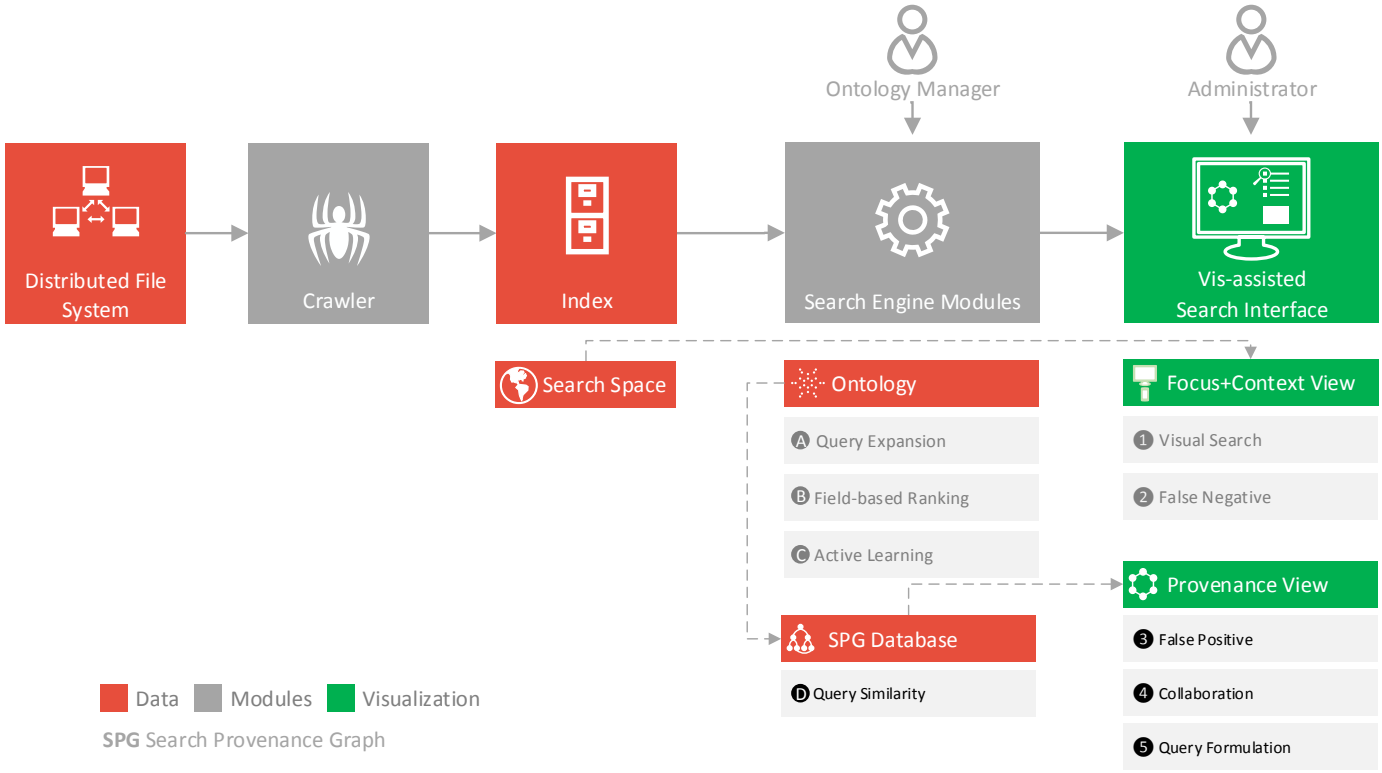


Figure 4: The architecture of a special-purpose enterprise search engine, where the provenance visualization facility is activated by the search interface, and retrieval of historical query information from the SPG Database. The SPG and Provenance View are the main focus of this paper. The Focus+Context View was proposed in [69].

gone through series of training in laboratory conditions. We implemented active learning mechanism for the ontologies to accommodate the dynamics in a real world environment captured from user relevance feedback information. The Ontologies can also be manually tuned by *Ontology Manager*.

The main visual and interactive features of the interface were implemented using AngularJS and the D3.js library [75]. The search engine modules were implemented in C++.

6. Computing Query Similarity for SPG

Query similarity defines the implicit connections in the SPG. In IR , session and textual features (e.g., TF*IDF, cosine similarity, Jaccard's index, character-level 3-grams and Levenshtein distance) have been used to compute query similarity (e.g., [32, 76, 77, 78]). Fast Algorithms for mining association rules as query similarity is reported in [79]. Among these, the TF-IDF and cosine similarity measure has been widely applied to content matching scenarios. These simple similarity measures can deliver reasonable accuracy. However, it is well understood that many do not include any semantic comparison. Problems arises when synonymies such as "building" and "construction" would be considered as not similar by these measures.

In our system, a search query consists of four fields: *keyword*, *type*, *size*, and *date*. The keyword field must contain at least one search term, while other fields can be set or left empty. In this work, we implemented a similarity function that computes the score based on all the four fields of the query. In particular, the keyword similarity is computed with the help of the knowledge base of the *enterprise search engine*.

6.1. Keyword Similarity

For measuring the similarity between two strings, we introduced a semantic similarity measurement supported by the *Term Ontology*, which is part of the multi-ontologies described in Section 5. We adapted the process of bipartite graph construction and computing an optimal bipartite graph described in [80, 81] and proposed an improved query similarity algorithm based on semantics. The modification is necessary in order to compare words that semantically related as defined in the Term Ontology. This algorithm follows two steps:

Step-1: Keyword Graph Construction. Given two queries Q_x and Q_y, we derive two sets $\mathbb{K}_x$ and $\mathbb{K}_y$, which contain the keywords or terms from the queries such that $\mathbb{K}_x \subset Q_x$ and $\mathbb{K}_y \subset Q_y$ Elements of $\mathbb{K}_x$ and $\mathbb{K}_y$ are considered as two disjoint sets and each element can be represented as a node to build a semantic connectivity between the queries. Each node in $\mathbb{K}_x$ is connected to every node of $\mathbb{K}_y$ in the Term Ontology, if there are no connection the distance is considered as ∞. By omitting the intermediate related nodes and considering only the path length between the nodes representing the original query terms, the semantic network can be considered to be a bipartite graph. The nodes of the first query and second query are the two vertex sets of the bipartite graph where the edge denotes the connection between the nodes as obtained from the semantic network.

The keyword graph is a complete weighted bipartite graph defined as $\Phi_K = \{\mathbb{K}_x, \mathbb{K}_y, \mathbb{W}\}$, where $\mathbb{W} \subseteq \mathbb{K}_x \times \mathbb{K}_y \times [0, 1]$ is a set of weighted edges. This semantic graph is used to compute the similarity between the queries.

For every pair of nodes, ($s_i \in \mathbb{K}_x$ and $d_j \in \mathbb{K}_y$), a function $\lambda : w \in [0, 1]$ computes the weight between s_i and d_j. For each node $s_i \in \mathbb{K}_x$ and $d_j \in \mathbb{K}_y$ we calculate the *shortest_path*(s_i, d_j) a set of weighted edges representing the shortest path between s_i and d_j in the *Term Ontology*. Let $w_1, w_2, \ldots, w_k$ be the weights on the k edges connecting s_i and d_j. The connectivity is determined by an aggregate weight between a source node s_i and a destination node d_j. The aggregated weight $\lambda(s_i, d_j)$ is computed as follows:

$$\lambda(s_i, d_j) = \prod_{l=1}^{k} w_l \rho(l)$$

where, $\rho(l)$ is a step function that moderates the impact of the weight according to connectivity between node n_l and edge l. A commonly-used step function is $\rho(l) = l^{\alpha}, \alpha \leq 0$. As $l \in [1, k]$ is an integer variable, the function appears to be stepping-down or horizontal (when $\alpha = 0$). We set $\alpha = 0$, i.e., $\rho(l) \equiv 1$, in this work.

Alternative step functions may also be used. For example, to compute $\lambda(s_i, d_j)$ using only the notion of synset in WordNet, one would have $\rho(l) = 1$ if $l = 1$ and $\rho(l) = 0$ otherwise. To consider all descendants which are k links away from s_i, one would have $\rho(l) = 1$ if $l \leq k$ and $\rho(l) = 0$ otherwise. To consider all descendants in the 'is-a' hierarchy, one would have $\rho(l) = 1$ if edges, $1, 2, \ldots, l$ are all "is-a" connections, and $\rho(l) = 0$ otherwise. The recursion stops when $\lambda(s_i, d_j)$ becomes insignificant, i.e., less than a threshold value. In this work, the threshold is set as 0.40.

Step-2: Bipartite Matching. Once the complete bipartite graph for the keywords is derived, we are able to apply standard algorithms for optimal matching of the bipartite graph. A matching $\mathbb{W}_{opt} \subseteq \mathbb{W}$ is a collection of weighted edges such that every vertex of $\mathbb{K}_x \cup \mathbb{K}_y$ is incident to at most one edge of $\mathbb{W}_{opt}$. A matching is perfect if its cardinality is equal, i.e., $|\mathbb{K}_x| = |\mathbb{K}_y|$. Here, the optimal bipartite matching algorithm operates under the simple notion that the connections with higher weights i.e., closely connected keyword nodes, contribute more to the similarity value.

Given the bipartite representation $\Phi_{K_{opt}} \subseteq \Phi_K$, the optimal matching between two keyword or vertex sets can be computed using a maximum allocation algorithm. An optimal maximum allocation algorithm can yield an improved performance, however, as the number of keywords in a search query is a small number, a brute-force algorithm performs reasonably well. The optimal bipartite graph, $\Phi_{K_{opt}} = \{\mathbb{K}_x, \mathbb{K}_y, \mathbb{W}_{opt}\}$ where, $\mathbb{W}_{opt} \subseteq \mathbb{W}$ such that $\Theta(\mathbb{K}_x, \mathbb{K}_y)$ is maximum, is computed as:

$$\forall w_i \in \mathbb{W}_{opt}, \;\; \Theta(\mathbb{K}_x, \mathbb{K}_y) = 1 - \prod_{i=1}^{|\mathbb{W}_{opt}|} (1 - w_i)$$

6.2. Similarity: Type, Size, and Date Fields

As mentioned earlier the value of the type, size, and date fields can be set or left empty. Therefore, we introduce two different metrics to compute the similarity of these fields, we call *Behavioral Similarity* and *Value Similarity*. Given two queries, the *Behavioral Similarity* function Ψ measures the similarity based on whether the fields are populated (set) or are empty (unset) and the *Value Similarity* function Ω measures the degree of match. They are defined as:

$$\Psi(a, b) = \begin{cases} 1 & \text{if } a \;\&\; b \text{ both are set} \\ 0.5 & \text{if both are unset} \\ 0 & \text{if one is set and another is unset} \end{cases}$$

$$\Omega(a, b) = \begin{cases} 1 & \text{if } a \;\&\; b \text{ are the same} \\ 0.5 & \text{if there is an overlap between } a \;\&\; b \\ 0 & \text{if there is no overlap} \end{cases}$$

6.3. Aggregate Score

Finally, a function Λ computes the similarity score between two queries Q_x and Q_y by combining the similarity scores of keywords ($\mathbb{K}_x$ and $\mathbb{K}_y$), types (t_x and t_y), sizes (s_x and s_y), and dates (d_x and d_y) fields.

$$\Lambda(Q_x, Q_y) = (1 - \alpha - \beta - \gamma)\, \Theta(\mathbb{K}_x, \mathbb{K}_y) + \alpha\, \Psi(t_x, t_y)\, \Omega(t_x, t_y) + \beta\, \Psi(s_x, s_y)\, \Omega(s_x, s_y) + \gamma\, \Psi(d_x, d_y)\, \Omega(d_x, d_y)$$

where $0 \leq (\alpha + \beta + \gamma) < 1$. α, β, and γ are scalar constants that defines the influence of type, size and date fields respectively in the similarity measurement. In this project, we set $\alpha = \beta = \gamma = 0.1$, which can be reconfigured.

7. Visualization of SPG

Although it is possible to use the basic visual representation of search provenance graph as shown in Figure 3, users have to use pop-up windows to retrieve information related previous queries. Frequent interactions can be expensive and, particularly, it is easy to forget what was just shown in the pop-up windows activated. We hence introduced glyphs to enrich each node in a SPG to facilitate a high-level IR at a glance as well as externalize the user's memory about each query.

We began our visual design activities with a series of brainstorming meetings. At first we identified the most commonly-used search query attributes (e.g., keyword/name, type, size, date and search result attributes) to be visualized. We then identified summary features of search results that can help a user refine the queries. These include the number of newly-found files, how many files were in the previous search, how many files in the previous search are no longer shown, and how many files have already been selected by the user. After several iterations and consultation with our industrial partner a suitable glyph design was chosen based on consideration of occlusion, clarity, visual separation, metaphoric abstraction for aiding learning and remembering, and encoding costs in terms of scalability.

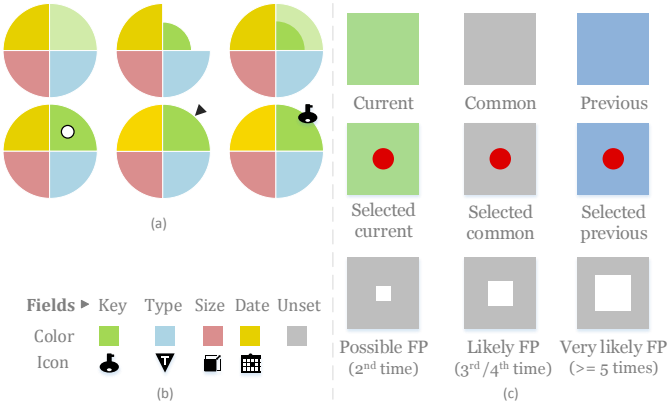


Figure 5: Glyph design for search provenance graphs. (a) Six different design options for characterizing a query, (b) the colors and icons used to highlight different query fields, and (c) the glyph components to depict a file in search results (FP: False Positive).

We designed two types of glyphs for representing search queries and search results respectively.

7.1. Glyphs for Search Queries

Six different design options for characterizing a query are presented in Figure 5 (a). A search query is composed of multiple fields and each field is related to a file system attribute, e.g., *name* or *keyword*, *type*, *size*, and *date*. During the query reformulation a user can set, unset, or change the values of any field. Hence the glyph design should ideally capture the temporal changes of the reformulated queries.

We began our design activities with a series of brainstorming meetings. We produced a table consisting of several optional visual channels for each attribute. After several iterations, a suitable option for each query field was identified on the basis of clarity, pop-out effect, visual separation, metaphoric abstraction for aiding learning and remembering, and encoding costs in terms of scalability. We decided to a k-wedge circle glyph to represent k search criteria. We used logological color encoding (e.g., **k**iwi color for **k**eyword, **s**carlet color for **s**ize, **t**urquoise for **t**ype, and **d**aisy for **d**ate). We considered five design options to indicate if a criterion has been added or changed in relation to a previous node as shown in Figure 5 (a). After consulting the industrial partner, we decided to use small pictograms to indicate such changes as they can also serve as a reminder about the name of the criterion concerned. These pictograms are shown in Figure 5 (b).

7.2. Glyphs for Search Results

A set of search results contains much complex information, and designing a summary glyph has been a challenging undertaking. We considered glyphs in the form of miniature treemaps, and found they cannot show much information. We thus decided to focus on some of the basic information needed that may aid the assessment of the effectiveness of a query, and stimulate hypotheses about better query reformulation. Such information includes (i) how many files are returned by the *current* query concerned, (ii) how many are *common* with the previous query, (iii) how many files returned by the *previous* query are no longer in this query, (iv) how many selected files (i.e., true positives) are among each of the above categories of files, and (v) how many files have been returned many times but were never selected (likely to be false positives). Once a file has met a search criteria but is not relevant to the user then this file is classified as *false positives*. When the same irrelevant file starts to appear in consecutive searches the false positive value of this file is incremented, see Figure 5 (c).

The basic glyph design for search results is a macro-glyph consisting of 25 cells, each is encoded with the above 5 pieces of information as shown in Figure 5 (c). Following an initial testing, we notice that many initial queries typically result in a large list of files. The 25 cells are not enough. We thus added another type of glyph for showing the overall statistics when there are more than 25 files in the three categories (current, common and previous).

7.3. An Example Use of the Provenance Visualization

This example visualizes the search provenance information of four administrators named Charlotte, Julie, Andy, and Jane who collaborated for four working days to create a portfolio of files related to a project for a customer.

7.3.1. SPG: Focus+Context View

Focus+Context view shows a subgraph of an SPG which is associated with a project. How one administrator's search (focus) relate to other administrators search (context) who are working collaboratively on a same project. Figure 6 shows an example focus+context graph where the four users mentioned earlier searched collaboratively for four working days.

Each search query is mapped to a node of the graph and shown as a dot with different color coding for different users. Each dot is positioned primarily according to the time when the search event took place. In the central line, we show the focus nodes, here Julie's search queries. The context nodes, here Charlotte, Andy, and Jane's search queries are shown above as well as below the central line. A link from a focus node to a corresponding context node is drawn such that if the context event has occurred earlier the link will connect to the node placed above else otherwise. This technique significantly reduce the occlusion and cluttering.

In this example, the first two days of searches were performed by Charlotte. It can be easily seen that the user is an inexperienced administrator and has performed many searches some of them consecutively in a short time span. In the following week, the search process was resumed by Julie and Andy who are experienced administrators. In the final day of search, Julie made few more searches, there after Jane joined and performed few queries and final quality checks.

We have used a linear time scale, however, the entire search session is divided into different slots to minimize the empty space when there are no search activity. If there are no search activities for more than an hour (this value can be configured) a glyph is placed to show the elapsed time as a session gap. If the elapsed time is in the range of hours we place a clock icon displaying the amount of hours, and if it is in days then a calendar icon showing the number of days elapsed. A time scale at the

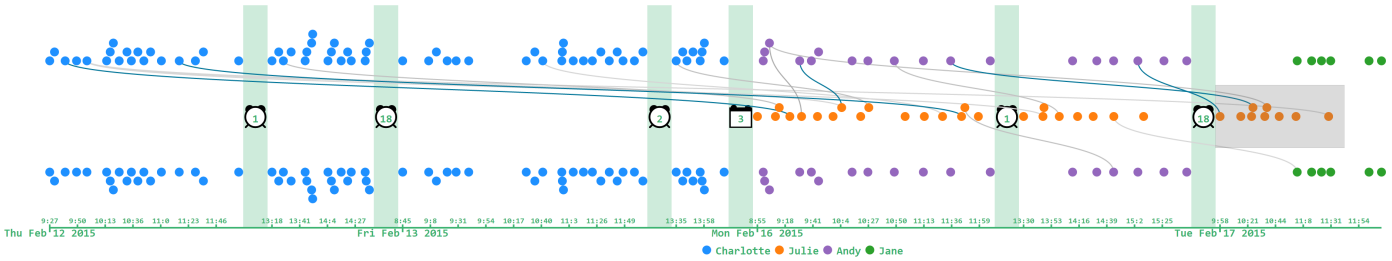


Figure 6: This example focus+context view of SPG displayed on Julie's search provenance window. It illustrates how Julie (focus) collaborated with other administrators (context). For example, Julie's fourth query is formulated based on Charlotte and in Julie's second day of search she formulated two queries based on Andy. The last nine queries from Julie's search are selected for glyph-based view as shown in Figure 7.



Figure 7: An example glyph-based visualization of the queries selected in Figure 6. The search queries are shown above the central line, and the search results are shown beneath the central line. Each search query may feature four criteria, clockwise, keyword, type, size and date. An icon appears when this criterion is new or has been changed. The search results are shown in two forms: summary statistics are shown when research results consist of more than 25 files and macro-glyph otherwise. For search results, green represents newly found files, gray represent files common to the current and previous queries, blue represents files appeared only in the previous query, and the red represents the selected files. The background color of the 'Late Discovery' dialog is gray as there is currently no newly discovered file.

bottom shows the dates (below) and time of day (above).

In order to scale the nodes in a given boundary one simple solution is to reduce the size of the dots based purely on the zoom-factor. This solution scales well as the dots can be seen even if they are very small. However, when the dots become too small to be understandable the placement algorithm determines the appropriate size and position of each dot in the time scale and shifts them vertically in order to maintain clarity while preventing serious cluttering and occlusion. The main drawback of this focus+context graph is that it does not show how the administrators searches relate to each other. Therefore, we introduced another visualization technique called *Collaboration Summary View*.

7.3.2. SPG: Glyph-based View

Figure 7 shows the provenance visualization where glyphs for search queries and search results are shown above and beneath the central line respectively. This example shows the last nine queries performed by Julie, selected from the Focus+context graph view shown in Figure 6. The user formulated the search query by selecting a query performed by Andy on the previous day. The search query is to find a group of building material files with a keyword "building".

The search query was automatically expanded by the *Query Processor* using the term ontology to include related keywords such as "construction", "manufacture", "fabricate" and so on. The search result returned a ranked list of many files and the user selected a few files from the top of the list. The first macro glyph for search results showed that all search results were new (100%) that is the query as well as the results are completely different from the last search the user performed on the previous day. The user selected few files from the search result. Note that the user can see the total number of files in the *Treemap-based focus+contex* or *Tabular view* as well as in the tooltip when the mouse is moved over the glyph, this work was reported in [69].

The user then added a type constraint to the search criteria to narrow down the search, and submitted the new query. The second query showed that no new files were returned, and the refined results consisted of 39% common files and 61% files from the previous search disappeared from the list. There were still a large number of files in the search result. The user again made a few selections of files from the top of the list.

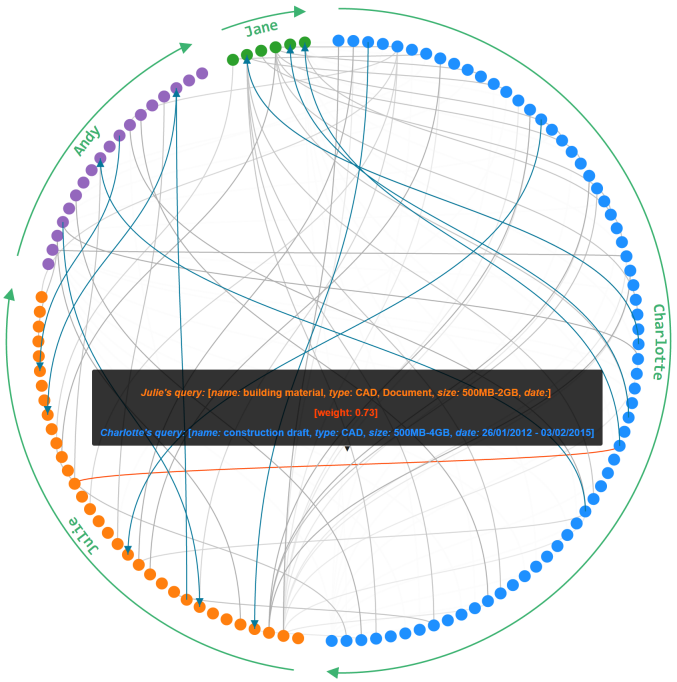


Figure 8: This example collaboration summary view of SPG illustrates the similarity between the search queries performed by the administrators associated with a project.

The user decided to add one more keyword, "material", in the query, and removed the type filter. The user also added a size filter particularly to exclude the files smaller than 500KB, and a date filter to exclude the files which are more than three years old. The new results consisted of 9% new files, 2% common and 89% files disappeared. The user continued with two further queries with reformulated search criteria. The reformulated query narrowed down the retrieved result significantly.

The user then decided to formulate a query based on the 2nd and 4th query. The user selected these two queries from the visualization window and specified a logical AND operation to combine their search criteria. The *Search Interface* reformulated a new query and populated the search form. The user was also given an option to review the query or modify it before submitting the new query. The 6th query found 8 new files while 13 files from the previous query now disappeared. The macro glyph changed to a cell-based representation as there were less than 25 files to show.

The user again reformulated the query based on the 6th query that resulted in few false positives and the first query that resulted in a large number of files, some of which were likely to be true positives but were not selected. This combination of search criteria resulted in four new files. The user continued to reformulate the query further. In the 9th query, there was no new file and six files were shown as potential false positives.

All these actions are stored in the *SPG Database*. At any time the user can save a portfolio of the selected files as well as resume the search process from any point.

7.3.3. SPG: Collaboration Summary View

The Collaboration Summary view of SPG uses a radial-based layout to visualize the relationships among the administrators searches working collaboratively on a project. The example summary graph shows the subgraph of an SPG which is associated with the project. The Collaboration Summary view allows the visualization of their queries relate to each other, and their explicit and implicit connections. The explicit connections are shown in pale-blue color, and the implicit or system computed similarities are mapped to translucent gray color. Figure 8 presents an example summary graph. Here, for example, Jane performed some last few searches as a quality check and her queries are mostly close to Charlotte's. It can also be seen that Andy and Julie did many collaborative and close searches. On selecting a link, a pop-up window shows the connected queries and the similarity weight.

In order to scale the nodes in a given boundary, similar to the focus+context graph, the size of the dots can be reduced based on the zoom-factor. When the dots become too small to be understandable the radius of the radial graph can be increased and scrollbars can be added if needed.

8. Evaluation

8.1. Requirement Analysis

At the beginning phase of the project, there were meetings with members of the industrial partner, where the existing document management system and the requirements of the document search system were discussed. Thereafter, the author of this thesis had a week long placement in the enterprise. During that placement, there were a series of meetings with the domain experts, and had the opportunities to familiarize with the existing document management systems, the distributed file system and several file repositories used by different groups across the organization. We discussed about the requirements, the use cases, and the technical challenges of the system. Based on the knowledge gained, the special purpose enterprise search engine to search for files stored across a large distributed system in the organization was developed.

8.2. Organization

After implementing the glyph-based view, we visited the industrial partner to evaluate this visualization facility, together with the latest version of the search engine. During the two-day visit, four domain experts and managers were consulted. They all have an extensive experience of construction management as well as in dealing with large number of documents. The evaluation criteria included (a) the SPG and the glyph design options, and the discriminating capacities of different visual channels, (b) metaphoric mappings for aiding learning and remembering, (c) encoding costs in terms of pixel resolution, (d) importance and suitability of provenance visualization for identifying true positives, false positives, and false negatives, and (e) the usability of the system from the enterprise's perspective.

We had another one-day visit to industry to evaluate the focus+context view and the collaboration summary view of SPG to evaluate if such provenance visualization can assist admin-

istrators to perform search tasks collaboratively and efficiently. This second evaluation also prompted two domain experts to carry out business cost-benefit analysis.

8.3. Procedure

We consulted with four domain experts and managers individually as per their convenience. The evaluations were conducted through demonstration of the system and through discussions about the above evaluation criteria. The meeting with each of the experts began with a 15-minute introductory presentation followed by some 20 minutes of open discussion. The discussions were recorded and photographs were taken with the permission of the domain experts. Although the domain experts were aware of the project and the purpose of the meeting, they were still informed of the overall goals of the system, application scenarios, and objectives of the meeting.

8.4. Discussion

During discussions, domain experts provided various feedbacks and comments related to the above evaluation criteria. The discussions were lively, and the domain experts were highly enthusiastic in offering their opinions and suggestions. In particular, domain experts asked questions about glyph designs, visual encoding, the search interface and use case scenarios. The idea of visualizing search provenance has been appreciated by all of them. The domain experts learned the visual encoding very quickly, and offered informed suggestions about various design aspects. These suggestions were accommodated into a revised design.

- One expert asked if more query fields are introduced how well it will scale. Therefore, a crush test with 12 query fields to demonstrate the discriminating capacity of various visual channels at different resolutions was showed. The author also explained that the pictograms can change to a small arrow head (glyph design option four) when scalability is an issue, and this option was welcomed by them. A tooltip while hovering over the glyphs to show the query in text has been added later as suggested by the domain experts.

- The domain experts expressed that among all the glyph design options the options which use size or opacity to highlight changes in field may confuse the users, therefore, such options were not included.

- Visualizing the statistics for a large number of search results and attributes was liked by the domain experts. However, they wanted to see the percentage as well as the number of files which are new, common, and deleted. The percentage values were showed in the macro glyphs for search results and a tooltip feature to show the exact number on mouse hover was introduced.

- In order to highlight the selected files, domain experts suggested using a larger and salient red dot, we improved this in our revised design.

- One domain expert asked if there is a way to reformulate the query by combining the previous queries. A feature to perform a logical AND operation was implemented to address this.

- The idea of showing false positives was seen as a useful feature. Such a file may be really important but duplicated, or it may be an outdated version. However, domain experts suggested to use an alternative design option to show the false positive, as the initial design option was to use a halo which was not intuitive and salient enough. In the revised version, the white square design was introduced.

- Domain experts emphasized that showing the explicit connections and implicit or system computed connections in SPG are extremely useful. Therefore, we have introduced different color coding to aid better visual discrimination capability.

The domain experts made a number of positive comments:

- The circular wedge-based design is preferred by all the domain experts over the alternative design options discussed. They found the choice of pictograms and colors to highlight different query fields intuitive as the pictograms relate to the name of the field.

- The idea of helping stimulating hypotheses about false negatives was much appreciated, as in a large organization most know that some documents are expected at certain locations or business units but not yet found.

- One domain expert expressed that training would be required to learn and memorize the search result glyphs. However, this was not considered to be an issue. Another domain expert mentioned that after using the system over a short period, users can easily remember the fields.

- The domain experts instantly grasped the functionality of the SPG and the visual representations. They found that the collaborative graphs are helpful to avoid repetitive work or learn from previous searches. As all the search actions are stored and can be visualized later on, the SPG can assist the lead administrator to perform performance review, assess work quality and train new users, commented by the experts. As sometimes one administrator might have to step into a project to resume a search from any point past, the focus+context view would allow one to overview the situation quickly.

- The domain experts found that the provenance visualization enables quick interpretation of the search criteria, and improved quality of the retrieved results.

- One domain expert said: “This is a nice and simple representation to show the high-level overview of the reformulated queries and the relationship among the results. Showing the search result size, and the selection within and comparing with the previous search is nice.” Another domain expert commented: “Showing the relationship in

terms of new, common and deleted results in the form of a glyph is a great idea, as it can guide the user by giving an idea to effectively reformulate a query."

- Another domain expert commented: "I find it quite useful, it makes sense, user can easily get an idea about whether they need to refine the results further, for example, if there are few new files in a search result the user may reformulate the query.

- Another expert commented: "It is a time consuming task to retrieve all the relevant documents related to a search. Many times during a search the user gets lost and performs the same search repeatedly. The use of this new visualization could eliminate this problem, and could therefore prove to be very useful. I can definitely see a business application of this search interface and visualization in an engineering enterprise where documents are so important in the same time very difficult to deal with it".

During the business cost estimation, we broke the search processes detailed action steps, including capture a file and place it in a server, enter a record in a database, database maintenance, search by a local administrator based in a project, search by central administrators with and without the aid of visualization, packaging selected search results, and so on.

After seeing the demo of the prototype system, we reassessed the cost analysis as described in Section 3.2. The experts in cost analysis became more confident about the cost model. The estimation of the costs show that using an enterprise search engine, one could save nearly 90% of time in comparison with using a database as the document retrieval mechanism. The bulk of savings would be from the effort for entering records about files into a database, though searches would be more accurate and quickly. However, when considering search effort only, visualization assisted search is expected to save some 30-50% of time in comparison to search without visualization. Therefore for a large construction enterprise, such a savings would not be trivial, particularly when it is translated into a monetary term.

The industrial partners realizes the important role and potential benefits of the visualization-assisted search solution to handle their large-scale data generated throughout different project life cycles which has been highlighted in their recently published magazine [82].

9. Conclusions

In this paper, we have presented a visualization-assisted search solution to support enterprise search. We have developed a prototype system featuring both an enterprise search engine and a set of visualization capabilities. The development of the provenance visualization has benefited particularly from the knowledge-base of the search engine. This demonstrates that visualization can and should play an integral role in an enterprise search engine. Our industrial feasibility study has shown that visualization can impact on industrial applications by improving working practices in the real world. The study helped the enterprise to realize the important role of visualization in search operations at an industrial scale. It provided the basis for establishing a cost-effective working practice, and developing new software for supporting such routine but intensive operations. From this work, we may draw a generalized conclusion that visualization can be used to help users identify false positives and reason about false negatives in search processes. Search provenance visualization provides an effective means for users to reformulate search criteria, facilitating external memorization in a search task that may be performed over a short period and by multiple users collaboratively.

As a sponsored research project, this feasibility study was not intended to deliver a piece of software to support the new BIM requirements in 2016, we believe that many design concepts produced in this work will be featured in future operational software. We are also in discussion with the sponsor about disseminating the source code of the search engine and visualization software.

Acknowledgements

The authors wish to thank Laing O'Rourke, UK for funding (project code DF11LO) this D.Phil. studentship of the first author. We are also grateful to Nick Lester, Claire Jones, Jasvir Mann, Florian Le Gouriellec, and Tes Adamou from Laing O'Rourke for sharing their knowledge and giving feedbacks during the placement and development of the system. We are thankful to Sohan Sangwan from HP, Bangalore for sharing his knowledge on AngularJS.

References

[1] D. Stenmark, A method for intranet search engine evaluations, in: Proc. IRIS22, 1999, pp. 1–14.

[2] M. Abrol, N. Latarche, U. Mahadevan, J. Mao, R. Mukherjee, P. Raghavan, M. Tourn, J. Wang, G. Zhang, Navigating large-scale semi-structured data in business portals, in: Proc. 27th Int. Conf. on VLDB, 2001, pp. 663–666.

[3] D. Hawking, Challenges in enterprise search, in: Proc. 15th Australasian Database Conf., Vol. 27, 2004, pp. 15–24.

[4] S. Feldman, C. Sherman, The High Cost of Not Finding Information Information: The Lifeblood of the Enterprise, in: IDC Technical Report (29127), 2003, pp. 1–9.

[5] G. Grefenstette, Enterprise search trends and challenges, in: CHORUS Final Conf., 2009.

[6] P. Dmitriev, P. Serdyukov, S. Chernov, Enterprise and Desktop Search, in: WWW, 2010, pp. 1345–1346.

[7] S. Cohen, C. Domshlak, N. Zwerdling, On Ranking Techniques for Desktop Search, ACM Trans. on Information Systems 26 (2) (2008) 1–24. doi:10.1145/1344411.1344417.

[8] O. Ben-Yitzhak, S. Yogev, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, Beyond basic faceted search, in: Proc. WSDM, 2008, p. 33. doi:10.1145/1341531.1341539.

[9] K. Balog, L. Azzopardi, M. de Rijke, Formal models for expert finding in enterprise corpora, in: Proc. SIGIR, 2006, p. 43. doi:10.1145/1148170.1148181.

[10] S. Allard, K. J. Levine, C. Tenopir, Design engineers and technical professionals at work: Observing information usage in the workplace, Journal of the American Society for Information Science and Technology 60 (3) (2009) 443–454. doi:10.1002/asi.v60:3.

[11] G. J. Hahm, J. H. Lee, H. W. Suh, Semantic relation based personalized ranking approach for engineering document retrieval, Advanced Engineering Informatics 29 (3) (2015) 366—-379. doi:10.1016/j.aei.2015.01.003.

[12] G. J. Hahm, M. Y. Yi, J. H. Lee, H. W. Suh, A personalized query expan-

sion approach for engineering document retrieval, Advanced Engineering Informatics 28 (4) (2014) 344–359. doi:10.1016/j.aei.2014.04.002.

[13] H.-T. Lin, N.-W. Chi, S.-H. Hsieh, A concept-based information retrieval approach for engineering domain-specific technical documents, Advanced Engineering Informatics 26 (2) (2012) 349–360. doi:10.1016/j.aei.2011.12.003.

[14] T. Montecchi, D. Russo, Y. Liu, Searching in Cooperative Patent Classification: Comparison between keyword and concept-based search, Advanced Engineering Informatics 27 (3) (2013) 335–345. doi:10.1016/j.aei.2013.02.002.

[15] O. Eck, D. Schaefer, A semantic file system for integrated product data management, Advanced Engineering Informatics 25 (2) (2011) 177–184. doi:10.1016/j.aei.2010.08.005.

[16] G. Gao, Y.-S. Liu, M. Wang, M. Gu, J.-H. Yong, A query expansion method for retrieving online BIM resources based on Industry Foundation Classes, Automation in Construction 56 (2015) 14–25. doi:10.1016/j.autcon.2015.04.006.

[17] G. Gao, Y.-S. Liu, P. Lin, M. Wang, M. Gu, J.-H. Yong, BIMTag: Concept-based automatic semantic annotation of online BIM product resources, Advanced Engineering Informatics-doi:10.1016/j.aei.2015.10.003.

[18] G. Gao, Y.-S. Liu, M. Wang, X.-G. Han, BIMTag: Semantic annotation of Web BIM product resources based on IFC ontology, in: 21st Int. Workshop: Intelligent Computing in Engineering (EG-ICE), 2014, pp. 1–11.
URL http://cgcad.thss.tsinghua.edu.cn/liuyushen/main/pdf/EGICE2014{_}BIMTag{_}final2.pdf

[19] M. Dörk, S. Carpendale, C. Collins, C. Williamson, VisGets: coordinated visualizations for web-based information exploration and discovery, IEEE Trans. on Visualization & Comp. Graphics 14 (6) (2008) 1205–12. doi:10.1109/TVCG.2008.175.

[20] L. Carata, S. Akoush, N. Balakrishnan, T. Bytheway, R. Sohan, M. Selter, A. Hopper, A primer on provenance, Communications of the ACM 57 (5) (2014) 52–60. doi:10.1145/2602649.2602651.

[21] S. P. Callahan, J. Freire, E. Santos, C. E. Scheideggers, C. T. Silva, H. T. Vo, VisTrails: Visualization Meets Data Management, in: Proc. SIGMOD, 2006, p. 745. doi:10.1145/1142473.1142574.

[22] UK Cabinet Office, Government Construction Strategy (2011).
URL www.cic.org.uk/gcs/GovernmentConstructionStrategy.pdf

[23] Lucene.
URL https://lucene.apache.org

[24] Solr.
URL http://lucene.apache.org/solr

[25] Indri.
URL http://www.lemurproject.org/indri

[26] S.-H. Hsieh, H.-T. Lin, N.-W. Chi, K.-W. Chou, K.-Y. Lin, Enabling the development of base domain ontology through extraction of knowledge from engineering domain handbooks, Advanced Engineering Informatics 25 (2) (2011) 288–296. doi:10.1016/j.aei.2010.08.004.

[27] M. Dörk, C. Williamson, S. Carpendale, Towards Visual Web Search : Interactive Query Formulation and Search Result Visualization, WWW Workshop on Web Search Result Summarization and Presentation (2009) 2–5.

[28] M. Dörk, C. Williamson, S. Carpendale, Navigating Tomorrow's Web: From Searching and Browsing to Visual Exploration, ACM Trans. on the Web 6 (3) (2012) 1–28. doi:10.1145/2344416.2344420.

[29] M. Dörk, S. Carpendale, C. Williamson, Fluid Views: A Zoomable Search Environment, in: Proc. Int. Working Conf. on Advanced Visual Interfaces, 2012, p. 233. doi:10.1145/2254556.2254599.

[30] M. Dork, N. H. Riche, G. Ramos, S. Dumais, PivotPaths: Strolling through Faceted Information Spaces, IEEE Trans. on Visualization & Comp. Graphics 18 (12) (2012) 2709–2718. doi:10.1109/TVCG.2012.252.

[31] A. Nocaj, U. Brandes, Organizing Search Results with a Reference Map, IEEE Trans. on Visualization & Comp. Graphics 18 (12) (2012) 2546–2555. doi:10.1109/TVCG.2012.250.

[32] E. Gomez-Nieto, F. San Roman, P. Pagliosa, W. Casaca, E. S. Helou, M. C. F. de Oliveira, L. G. Nonato, Similarity preserving snippet-based visualization of web search results., IEEE Trans. Visualization & Comp. Graphics 20 (3) (2014) 457–70. doi:10.1109/TVCG.2013.242.

[33] E. Isaacs, K. Domico, S. Ahern, E. Bart, M. Singhal, Footprints: A Visual Search Tool that Supports Discovery and Coverage Tracking, IEEE Trans. on Visualization & Comp. Graphics 20 (12) (2014) 1793–1802. doi:10.1109/TVCG.2014.2346743.

[34] L. Bavoil, S. P. Callahan, C. E. Scheidegger, P. J. Crossno, C. T. Silva, J. Freire, H. T. Vo, VisTrails: Enabling Interactive Multiple-View Visualizations, in: IEEE VIS, 2005, pp. 135–142. doi:10.1109/VISUAL.2005.1532788.

[35] C. Scheidegger, D. Koop, E. Santos, H. Vo, S. Callahan, J. Freire, C. Silva, Tackling the Provenance Challenge one layer at a time, Concurrency and Computation: Practice and Experience 20 (5) (2008) 473–483. doi:10.1002/cpe.v20:5.

[36] O. Biton, S. Cohen-Boulakia, S. B. Davidson, C. S. Hara, Querying and Managing Provenance through User Views in Scientific Workflows, in: IEEE 24th Int. Conf. on Data Engineering, 2008, pp. 1072–1081. doi:10.1109/ICDE.2008.4497516.

[37] E. Maguire, P. Rocca-Serra, S.-A. Sansone, J. Davies, M. Chen, Taxonomy-based Glyph Design – with a Case Study on Visualizing Workflows of Biological Experiments, IEEE Trans. on Visualization & Comp. Graphics 18 (12) (2012) 2603–2612. doi:10.1109/TVCG.2012.271.

[38] E. Maguire, P. Rocca-Serra, S.-A. Sansone, J. Davies, Visual Compression of Workflow Visualizations with Automated Detection of Macro Motifs, IEEE Trans. on Visualization & Comp. Graphics 19 (12) (2013) 2576–2585. doi:10.1109/TVCG.2013.225.

[39] D. Huynh, D. R. Karger, D. Quan, V. Sinha, Haystack: A Platform for Creating, Organizing and Visualizing Semistructured Information, in: Proc. 8th Int. Conf. on Intelligent User Interfaces, 2003, p. 323. doi:10.1145/604045.604116.

[40] N. Del Rio, P. P. Da Silva, Probe-it! Visualization Support for Provenance, in: Int. Conf. on Advances in Visual Computing, 2007, pp. 732–741. doi:10.1007/978-3-540-76856-2_72.

[41] P. Macko, S. Margo, Provenance map orbiter: Interactive exploration of large provenance graphs., in: TaPP, 2011, pp. 1–6.

[42] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, G. Robertson, GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks while Supporting Exploration History, in: Proc. CHI, 2012, p. 1663. doi:10.1145/2207676.2208293.

[43] M. A. Borkin, C. S. Yeh, M. Boyd, P. Macko, K. Z. Gajos, M. Seltzer, H. Pfister, Evaluation of filesystem provenance visualization tools, IEEE Trans. on Visualization & Comp. Graphics 19 (12) (2013) 2476–85. doi:10.1109/TVCG.2013.155.

[44] R. Walker, A. Slingsby, J. Dykes, K. Xu, J. Wood, P. H. Nguyen, D. Stephens, B. L. W. Wong, Y. Zheng, An extensible framework for provenance in human terrain visual analytics., IEEE Trans. on Visualization & Comp. Graphics 19 (12) (2013) 2139–48. doi:10.1109/TVCG.2013.132.

[45] J. Zhang, C. Chen, M. S. Vogeley, D. Pan, A. Thakar, J. Raddick, SDSS Log Viewer: visual exploratory analysis of large-volume SQL log data, in: Proc. Visualization and Data Analysis, Vol. 8294, 2012, p. 13. doi:10.1117/12.907097.

[46] M. Gray, A. Badre, M. Guzdial, Visualizing usability log data, in: Proc. IEEE Symp. on Information Visualization, 1996, pp. 93–98. doi:10.1109/INFVIS.1996.559227.

[47] J. Wei, Z. Shen, N. Sundaresan, K.-L. Ma, Visual cluster exploration of web clickstream data, in: IEEE VAST, 2012, pp. 3–12. doi:10.1109/VAST.2012.6400494.

[48] S. K. Card, P. Pirolli, M. Van Der Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, J. Boshart, Information scent as a driver of web behavior graphs: results of a protocol analysis method for web usability, in: Proc. SIGCHI, 2001, pp. 498–505. doi:10.1145/365024.365331.

[49] J. Lee, M. Podlaseck, E. Schonberg, R. Hoch, Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising, Data Mining and Knowledge Discovery 5 (1-2) (2001) 59–84. doi:10.1023/A:1009843912662.

[50] E. H. Chi, Improving Web usability through visualization, IEEE Internet Computing 6 (2) (2002) 64–71. doi:10.1109/4236.991445.

[51] S. Waterson, J. I. Hong, T. Sohn, J. Heer, T. Matthews, J. Landay, What Did They Do? Understanding Clickstreams with the WebQuilt Visualization System, in: Proc. Working Conf. on Advanced Visual Interfaces, 2002, pp. 94–102. doi:10.1145/1556262.1556276.

[52] J. E. Pitkow, K. A. Bharat, WEBVIZ: A Tool for World Wide Web Access

Log Analysis, in: Proc. WWW, 1994, pp. 271–277.
[53] Z. Shen, J. Wei, N. Sundaresan, K.-L. Ma, Visual analysis of massive web session data, in: IEEE Symp. on Large Data Analysis and Visualization, 2012, pp. 65–72. doi:10.1109/LDAV.2012.6378977.
[54] D. R. Heidi Lam, D. Tang, T. Munzner, Session Viewer: Visual Exploratory Analysis of Web Session Logs, in: IEEE Symp. on Visual Analytics Science and Technology, 2007, pp. 147–154. doi:10.1109/VAST.2007.4389008.
[55] C. Shi, Y. Wu, S. Liu, H. Zhou, H. Qu, LoyalTracker: Visualizing Loyalty Dynamics in Search Engines, IEEE Trans. on Visualization & Comp. Graphics PP (99) (2014) 10. doi:10.1109/TVCG.2014.2346912.
[56] S. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, D. C. Robbins, Stuff I've seen: a system for personal information retrieval and re-use, in: Proc. SIGIR, 2003, pp. 72–79. doi:10.1145/860435.860451.
[57] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, E. Giannopoulou, Ontology visualization methods—a survey, ACM Computing Surveys 39 (4) (2007) 10–es. doi:10.1145/1287620.1287621.
[58] D. Duke, K. Brodlie, D. Duce, I. Herman, Do You See What I Mean?, IEEE Comp. Graphics and Applications 25 (3) (2005) 6–9. doi:10.1109/MCG.2005.55.
[59] P. Rhodes, E. Kraemer, B. Reed, VisIOn: an interactive visualization ontology, in: Proc. 44th Annual Southeast Regional Conf., ACM, 2006, pp. 405–410. doi:10.1145/1185448.1185538.
[60] O. Gilson, N. Silva, P. Grant, M. Chen, From Web Data to Visualization via Ontology Mapping, Comp. Graphics Forum 27 (3) (2008) 959–966. doi:10.1111/j.1467-8659.2008.01230.x.
[61] S. Carpendale, M. Chen, D. Evanko, N. Gehlenborg, C. Gorg, L. Hunter, F. Rowland, M.-A. Storey, H. Strobelt, Ontologies in biological data visualization, IEEE Comp. Graphics and Applications 34 (2) (2014) 8–15. doi:10.1109/MCG.2014.33.
[62] M. E. Dolan, J. A. Blake, Using ontology visualization to facilitate access to knowledge about human disease genes, Applied Ontology 4 (1) (2009) 35–49. doi:10.3233/AO-2009-0064.
[63] T. Hauer, D. Rogulin, S. Zillner, A. Branson, J. Shamdasani, A. Tsymbal, M. Huber, T. Solomonides, R. Mcclatchey, An Architecture for Semantic Navigation and Reasoning with Patient Data - Experiences of the Health-e-Child Project, in: Proc. 7th Int. Conf. on The Semantic Web, Vol. 5318, 2008, pp. 737–750. doi:10.1007/978-3-540-88564-1.
[64] C. Andronis, A. Sharma, V. Virvilis, S. Deftereos, A. Persidis, Literature mining, ontologies and information visualization for drug repurposing., Briefings in bioinformatics 12 (4) (2011) 357–68. doi:10.1093/bib/bbr005.
[65] A. Soylu, M. G. Skjæveland, M. Giese, I. Horrocks, E. Jimenez-Ruiz, E. Kharlamov, D. Zheleznyakov, A Preliminary Approach on Ontology-Based Visual Query Formulation for Big Data, in: Metadata and Semantics Research, Vol. 390, Springer, 2013, pp. 201–212. doi:10.1007/978-3-319-03437-9.
[66] Y. Tu, H.-W. Shen, GraphCharter: Combining browsing with query to explore large semantic graphs, in: IEEE Pacific Visualization Symp., 2013, pp. 49–56. doi:10.1109/PacificVis.2013.6596127.
[67] X. Ma, E. J. M. Carranza, C. Wu, F. D. van der Meer, Ontology-aided annotation, visualization, and generalization of geological time-scale information from online geological map services, Computers & Geosciences 40 (2012) 107–119. doi:10.1016/j.cageo.2011.07.018.
[68] A. Watson, Digital buildings - Challenges and opportunities, Advanced Engineering Informatics 25 (4) (2011) 573–581. doi:10.1016/j.aei.2011.07.003.
[69] S. Khan, K. J. Proctor, S. Walton, R. Bañares-Alcántara, M. Chen, A Study on Glyph-based Visualisation with Dense Visual Context, in: Computer Graphics and Visual Computing (CGVC), The Eurographics Association, 2014, pp. 73–80. doi:10.2312/cgvc.20141212.
[70] M. Chen, A. Golan, What May Visualization Processes Optimize?, IEEE Trans. on Visualization & Comp. Graphics (PrePrints) (2015) 1–14. doi:10.1109/TVCG.2015.2513410.
[71] M. Chen, L. Floridi, R. Borgo, What Is Visualization Really For?, in: The Philosophy of Information Quality, Springer Synthese Library, 2014, pp. 75–93. doi:10.1007/978-3-319-07121-3.
[72] R. Baeza-Yates, Graphs from Search Engine Queries, in: Proc. 33rd Conf. on Current Trends in Theory and Practice of Comp. Sc., Vol. 4362 of LNCS, 2007, pp. 1–8. doi:10.1007/978-3-540-69507-3.
[73] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna, The Query-flow Graph: Model and Applications, in: Proc. CIKM, 2008, pp. 609–618. doi:10.1145/1458082.1458163.
[74] G. A. Miller, WordNet: A Lexical Database for English, Communications of the ACM 38 (11) (1995) 39–41. doi:10.1145/219717.219748.
[75] M. Bostock, V. Ogievetsky, J. Heer, D3: Data-Driven Documents, IEEE Trans. Visualization & Comp. Graphics 17 (12) (2011) 2301–2309. doi:10.1109/TVCG.2011.185.
[76] P. Boldi, F. Bonchi, C. Castillo, S. Vigna, Query reformulation mining: models, patterns, and applications, Information Retrieval 14 (3) (2010) 257–289. doi:10.1007/s10791-010-9155-3.
[77] I. Bordino, C. Castillo, D. Donato, A. Gionis, Query similarity by projecting the query-flow graph, in: Proc. SIGIR, 2010, p. 515. doi:10.1145/1835449.1835536.
[78] Z. Zhang, O. Nasraoui, Mining search engine query logs for query recommendation, in: Proc. WWW, 2006, p. 1039. doi:10.1145/1135777.1136004.
[79] B. M. Fonseca, P. B. Golgher, E. S. de Moura, N. Ziviani, Using Association Rules to Discover Search Engines Related Queries, in: Proc. 1st Conf. on Latin American Web Congress, IEEE Comp. Society, 2003, p. 66. doi:10.1109/LAWEB.2003.1250284.
[80] R. Thiagarajan, G. Manjunath, M. Stumptner, Computing semantic similarity using ontologies, in: Int. Semantic Web Conf., 2008, pp. 1–16.
[81] A. Rajaraman, J. D. Ullman, Mining of Massive Datasets, Cambridge University Press, 2011.
[82] Laing O'Rourke, Unlocking Digital Engineering, Engineering Excellence Journal (2014) 84.