The Case for Browser Provenance

Daniel W. Margo and Margo Seltzer
Harvard School of Engineering and Applied Sciences

Abstract

In our increasingly networked world, web browsers are important applications. Originally an interface tool for accessing distributed documents, browsers have become ubiquitous, incorporating a significant portion of user interaction. A modern browser now also reads email, plays media, edits documents, and runs applications. Consequently, browsers process large quantities of data, and must record metadata, such as history, to help users manage their data. Most of the metadata that modern browsers record is actually provenance – metadata that captures the causality and lineage of data obtained via the browser. We demonstrate that characterizing browser metadata as provenance and then applying techniques from the provenance research community enables new browser functionality. For example, provenance can improve both history and web search by indicating contextual and personal relationships between data items. Users can also answer complex questions about the origins of their data by querying provenance. Our initial results suggest these features are feasible to implement and could perform well in modern browsers.

1 Introduction

Web applications have made the web browser the most important application software for millions of users. Many of a user's documents are either obtained through the browser or exist solely on the web. For users, this creates data management problems that are analogous to similar problems encountered in file systems. Users frequently need to know, "Where did my stuff go?" or, "Where did all this stuff come from?"

File systems address these problems with features such as desktop search and shortcuts. Browsers offer similar features such as history search, bookmarks, saved passwords, and autocompletion For example, a major feature in Mozilla Firefox 3 is its "smart location bar" [5], a history search-based autocompletion Similarly, Google Chrome's "New Tab" page [1] presents history and history search as its most important features.

History search and similar features rely on history metadata that the browser records as users browse the web. For example, all browsers record a history of URLs they have visited. Fundamentally, this metadata describes actions and their consequences; when the user navigates through a series of pages, enters a password, and downloads a file, the browser's history describes these events and how they are related. This metadata is provenance – it describes how the browser state came to be or, if properly stored and queried, precisely how a downloaded object came to be. To the best of our knowledge, the implications and benefits of characterizing browser history as provenance is an unresearched area.

In this paper we explore browser history as provenance. We present use cases in history search, web search, and download management that browser provenance can address. We identify common actions in modern browsers and the provenance those actions generate, useful provenance that browsers do not store, and useful provenance algorithms that browsers do not apply. Provenance helps a browser answer questions such as, "Where did all this stuff come from?" by providing a natural way to store and query data lineage. Similarly, a browser can answer, "Where did my stuff go?" by improving its history and web search with contextual and personal relationships extracted from its provenance store. In short, browsers can benefit a great deal by characterizing their history as provenance.

This paper is structured as follows. Section 2 presents four use cases in history search, web search, and download management that browser provenance can address. Section 3 dissects the architecture of a modern browser's history, relates it to provenance, and identifies some of the challenges of doing so. Section 4 describes our current and future work, and we conclude in section 5.

2 Use Cases

2.1 Contextual History Search

Scenario: Suppose a user searches the web for "rosebud" and then navigates to a result, "Citizen Kane". Later, when she searches her history for rosebud, she expects this history search to return Citizen Kane, because she found Citizen Kane with that search term.

Currently: A browser with textual history search will return the web search page for rosebud, because that page contains the search term in both its title and URL. However, it will not return Citizen Kane, because it does not recognize there is a connection between rosebud and Citizen Kane.

With Provenance: Browser provenance would show that Citizen Kane descends from the search term rosebud. Therefore, a provenance-aware browser could evaluate and return Citizen Kane in its history search results. Shah et al. [13] implemented a provenance-based desktop search algorithm that is readily extensible to history search. Briefly, the algorithm performs a textual search and then reorders results by the relevance of their provenance neighbors. As a first-generation descendant of the rosebud web search page, Citizen Kane would receive substantial weight in such a search.

2.2 Personalizing Web Search

Scenario: In our earlier example, the search term "rosebud" described a sled, but suppose the user is a gardener. To her, rosebud describes a flower, and when she searches the web for rosebud she is frustrated by references to Citizen Kane.

Currently: The browser knows that the user often visits pages containing the words "flower", "gardening", etc. in their title or URL. However, unless many of those descriptors contain both rosebud and flower together, the browser does not capture the user's connection between rosebud and flower.

With Provenance: As described in section 2.1, a provenance-aware browser could see not just textual, but contextual relationships between rosebud and flower pages. The browser would be much more capable of recognizing this connection, and if it did it could supplement a rosebud web search with flower as an additional search term. More generally, there are many advanced operators in modern search engines that are intended for power users [3]. A provenance-aware browser could leverage these operators automatically for regular users.

One exciting implication of this approach is that the browser could personalize search results without giving information about the user to the search engine. The search engine would only see a search for "rosebud flower"; it would not know anything about the user's history. Conversely, existing web personalization techniques require the user to submit personal information to a third party and can only personalize services that share data with that party.

2.3 Time-Contextual History Search

Scenario: Suppose the user is a wine enthusiast. She wants to find a bottle of wine that she saw on a web page, but she is frustrated because a search against her history for "wine" returns many results. The problem is that she does not remember the details of the specific page. She does remember that she was also searching for plane tickets at the time.

Currently: Neither history nor web search can help the user here. Although current web searches excel at delivering the user to a canonical and popular wine page, discovering the search terms to produce a specific page can be an arcane task. Furthermore, these terms can change as the search engine aggregates new content.

With Provenance: A browser could record provenance that captures the relationship between pages viewed within a similar time span. To users, these associations are relevant: a study by Blanc-Brude and Scapin [7] shows that users almost always recall events associated with documents. A history search for "wine associated with plane tickets" is both natural to the user and likely to return the desired result. In fact, Gyllstrom and Soules [9] implemented a desktop search system on this premise.

2.4 Download Lineage

Scenario: A user can be tricked by an adversary into downloading malicious software. When the user discovers the infection, she will want to know how she was infected. If the user identifies the source of the infection, then she knows to avoid that page in the future, and can take actions such as notifying the web host. Alternatively, a user may want to know where a file came from for the purposes of source attribution or obtaining an up-to-date version.

Currently: Most browsers record downloads somewhere and can find the URL of a download; however, in many cases the URL is not informative. If the user does not recognize the URL, then she will ask, "How did I get to this URL?" Similarly, an image file may have come from some image hosting site, but this is not useful for author attribution. Thus, the user is forced to recursively search her history and perform forensics until she finds a page that she recognizes.

With Provenance: What the user really wants is, starting from a known location, the sequence of actions that resulted in the download – that is, the lineage of the download. In a provenance-aware browser, the solution is a path query: “Find the first ancestor of this file that the user is likely to recognize.” “Likely to recognize” can be defined in terms of history, e.g., the number of visits the user has made to the page.

Provenance path queries can also enable new functionality. For example, if the user decides a page is untrusted, she may then want to find all downloads descending from that page and check them for viruses. This might be difficult for a user doing forensics, but is a simple query: “Find all descendants of this page that are downloads”.

3 Browser History Architecture

Web browsers differ in their details, but conceptually all browsers are fairly similar. In our research, we focused on Mozilla Firefox 3 [2], because it is open source, relatively popular, and recently underwent a major revision of its history implementation [4]. However, the concepts we discuss (links, tabs, etc.) are common to most browsers, and we believe our research is widely applicable.

The fundamental objects of browser history are web pages. Every browser records visited pages and associates metadata with them, such as the frequency of visits. One type of metadata Firefox records is the HTTP referrer, the page that sent the browser to a particular page. The referrer is useful for many purposes, such as identifying and eliminating redirects from history search results.

The referrer establishes an implicit provenance relationship between the referring page and the target page. Other history metadata can establish similar relationships. For example, Firefox stores a table of “transitions”, the actions that load a particular page. Transitions are a superset of the referrer; they include actions such as the user clicking a bookmark or the relationship between top-level and embedded page content. Researchers such as Roussel et al. [12] have identified many other forms of history metadata that are useful to both users and developers of interfaces and data management systems.

Any browser’s history can be represented as a graph in which pages are nodes, relationships are edges, and both nodes and edges can have attributes. This graph can be reasonably large; one author’s history has accumulated more than 25,000 nodes over the past 79 days. Given the ubiquity of graph data in browsers, one might imagine that graph algorithms and queries would be similarly ubiquitous, but surprisingly this is not the case. To the best of our knowledge, there are no graph algorithms applied to the history in any modern browser.

However, this is not the only context in which the scientific community considers graphs of web pages and their relationships. Web search is often characterized as a graph problem, and many web search algorithms, such as Kleinberg’s HITS [10], are graph algorithms that exploit the relationships between pages. However, browser history differs from a typical web graph in a number of important ways. First, the browser collects metadata that a web crawler cannot, such as which links are actually traversed by users, as opposed to those that simply exist. Secondly, the browser collects metadata from its user, as opposed to crawling web content. Therefore, features premised on browser history are inherently personalized. The browser can and should be better than web search at answering queries such as, “Where is that page *I visited last month?*”

Finally, the metadata collected by the browser is provenance. Every relationship in the browser history corresponds to an action taken by the browser to obtain one set of data from another. When the user clicks a bookmark to obtain a page, or a top-level page loads some embedded content, the logs of these events are provenance records. By characterizing browser history as provenance, we open up a new field of solutions to browser data management problems.

We present the following taxonomy of browser provenance. This taxonomy is neither complete nor definitive; however, we have identified many interesting research questions and propose that it serve as a basis for future discussion.

3.1 Page Visits and Link Traversals

Pages and links are the foundations of a browser’s history graph. Characterizing history as provenance requires that we first address the problem that pages and links are not necessarily acyclic (whereas provenance, by definition, is). For example, if the user traverses from a search page to another page and then follows a link back to the search page, a cycle is created. A cycle implies that a new version of some object in the cycle must be created, e.g., a new page visit instance of the search page. This versioning scheme breaks cycles and allows us to treat pages and links in browser history as a directed acyclic graph. This problem bears a resemblance to provenance cycle detection and avoidance as performed by the PASS project prototype [11].

Browsers often implicitly version pages by including time stamps in the metadata associated with page visits. In addition to solving the cycle problem, this facilitates time-based queries. However, the storage of a versioned graph is non-trivial and introduces interesting design decisions. For example, are time stamps a property of pages or links? Versioning nodes (pages) is a common cycle-breaking technique and is used by PASS. However, time stamping edges (links) can also break cycles by creating a traversal order among edges.

Are time stamps attributes of objects, or does each version create a new instance of an object? If they are attributes, then the object instances are semi-structured and more difficult to store. If each version creates a new instance of the object, then it is more difficult to make queries over all the objects that describe a given page or link. Firefox stores its time stamps as instances of link traversals, because in Firefox general page queries are more common than link queries. However, this can make it difficult to run link queries and by extension graph algorithms, because many records of a given link traversal may exist.

There has been a considerable amount of provenance research on efficient storage and query. For example, Chapman et. al [8] developed general factorization and inheritance-based methods that are almost certainly applicable to browser history. However, there are also many interesting properties of the history graph that may lead to unique storage methods. For example, if both pages and links are versioned as new instances, and only link relationships are considered, the result is a tree structure. There were a number of early efforts by researchers such as Ayers and Stasko [6] to develop an interface that used this propery to visualize recent history; we believe it could also be used for efficient storage and query. We are interested in history graph storage as both an enabler of more powerful history queries and a novel provenance storage problem.

3.2 Other Relationships

Links are merely one type of page relationship. Other relationship-generating actions include typing in the location bar, opening a new tab, or clicking a bookmark. Compared to links, browsers treat these other relationships as second-class citizens. For example, when the user moves from page to page by typing in the location bar, most browsers will not record a relationship between the previous page and the new page. So ironically, if a user often takes advantage of advanced navigation features such as Firefox's "smart location bar," she will generate sparsely connected metadata.

Similarly, most browsers do not capture the time relationship between pages that are open simultaneously. Firefox time stamps page visits, but it does not time stamp other display-altering actions, such as page "close". Consequently, it is impossible to determine whether two pages were open simultaneously; from the perspective of Firefox history, every page is always open. The simple addition of a corresponding close to each page visit enables queries on time relationships. These relationships can be used by contextual searches, as described in sections 3.1 through 3.3. Conceptually, time relationships are undirected; however, when necessary they can be directed with an arbitrary time-ordering rule such as, "the first node opened in a time span points to later nodes."

Redirects and inner content are a special case; although they are link-like relationships, unlike other edges they are not generated as the result of a user action. They are relevant to many queries such as Data Lineage, but personalization algorithms may wish to exclude or otherwise ignore them. One approach such algorithms can take is to unify edges by ignoring nodes from which a redirect or inner content link occurs.

3.3 Other Nodes

Introducing new relationships into the browser history also introduces new nodes. If clicking a bookmark generates a provenance relationship, then bookmarks must exist as nodes in the provenance store. Similarly, downloads and search terms can be represented as history nodes. Most browsers record these objects, but do not consider them a part of the browser's history graph. For example, querying a bookmark relationship may require the user to join heterogeneous tables or even databases in order to connect the bookmark store with the history store.

Search terms and form history are particularly useful provenance nodes. They are concise, conceptual, user-generated descriptors that are in the lineage of the page they generate and that page's descendants. When the user searches her history database, at the very least she expects it to return any page in her history that would also show up in a web search. Currently, this does not happen; but if search terms are stored as provenance, a contextual search can retrieve and use them as data descriptors. Furthermore, forms and form-generated pages are "deep web" content that are considered difficult for traditional search engines to capture and index. But they are easy to capture from the browser; in fact, many browsers already capture form history to provide autocomplete features.

3.4 Summary: Our Vision

Our idealized vision of browser metadata is a single, homogeneous provenance graph store that describes and relates every kind of history object. Efficient graph storage techniques permit relationships, paths, and neighborhoods to be queried with the same power as node objects. Provenance relationships such as bookmark creation and searches are stored and queried in the same manner as traditional page-link relationships. Users and algorithms do not have to connect heterogeneous data sets to explore the relationships between different kinds of history objects.

4 Implementation and Future Work

We have implemented a model browser provenance schema based on the Firefox Places [4] schema as a SQLite relational database. This schema stores heterogeneous provenance objects (such as pages and bookmarks) as homogeneous graph nodes. The total storage overhead of this schema over Places is 39.5%, but on real data, this represents less than 5MB because Places is quite conservative. Using this schema, we have implemented basic queries for all of our use cases and begun evaluating them on a real user history of over 25,000 nodes. From an information retrieval standpoint, these queries are fairly naive; our purpose at this time is not to find the best algorithms for browser provenance, but rather to show such algorithms are feasible for browsers to compute locally.

We implement "Contextual History Search" as a graph neighborhood expansion algorithm, similar to web search algorithms such as Kleinberg's HITS [10]. "Personalizing Web Search" performs term frequency analysis on the results of a contextual history search to find terms in user history associated with the search term. "Time-Contextual History Search" is a query over time relationships, and "Download Lineage" is a breadth-first search over a node's ancestors. These queries complete in less than 200ms in the majority of cases and can be bound to that time in the remaining cases. The challenges we encountered in implementing these queries and the details of our results will be published in future work.

Our initial results suggest that interesting graph algorithms on browser metadata are feasible for browsers to compute locally. However, there is still much work to be done. We must now develop more intelligent algorithms that can respond to our use case queries with high-quality results. There is a great deal of existing information retrieval research on web search on which we can build, but we also believe that our assumptions about browser provenance must be different from those of the web and that there are unique properties of browser provenance graphs we can exploit.

Another important issue for future work that we have not yet discussed is privacy. Browser history potentially contains a great deal of sensitive personal data. Techniques that aggregate this data at centralized locations and third parties can be powerful, but they must also answer difficult questions about the anonymity and privacy of their users. In our current and future work, the approach we take is to use browser provenance to increase user privacy by processing the data on the user's machine. We believe there is much more interesting research to be done with regards to provenance-based user-side personalization features.

5 Conclusion

This paper characterizes and connects browser history metadata to provenance. It describes use cases in history search, web search, and data management that provenance can address. It identifies useful provenance that browsers can capture and query and some of the challenges of doing so. We have begun implementing some of these solutions and believe there are many opportunities and challenges for future research, especially with regards to algorithm refinement, privacy, and personalization.

References

[1] Explore Google Chrome Features: New Tab page. `http://www.google.com/support/chrome/bin/answer.py?answer=95451\&hl=en`, December 2008.

[2] Firefox web browser — Faster, more secure, & customizable. `http://www.mozilla.com/en-US/firefox/`, December 2008.

[3] Google Help: Cheat Sheet. `http://www.google.com/help/cheatsheet.html`, December 2008.

[4] Places - MDC. `https://developer.mozilla.org/en/Places`, December 2008.

[5] Smart Location Bar. `http://support.mozilla.com/en-US/kb/Smart+Location+Bar`, December 2008.

[6] AYERS, E. Z., AND STASKO, J. T. Using graphic history in browsing the world wide web. In *The 4th International World Wide Web Conference* (December 1995), pp. 11–14.

[7] BLANC-BRUDE, T., AND SCAPIN, D. What do People Recall about their Documents? Implications for Desktop Search Tools. In *Intelligent User Interfaces* (January 2007).

[8] CHAPMAN, A., JAGADISH, H., AND RAMANAN, P. Efficient Provenance Storage. In *SIGMOD* (June 2008).

[9] GYLLSTROM, K., AND SOULES, C. Seeing is retrieving: Building information context from what the user sees. In *Intelligent User Interfaces* (January 2008).

[10] KLEINBERG, J. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM 46* (September 1999).

[11] MUNISWAMY-REDDY, K.-K., HOLLAND, D., BRAUN, U., AND SELTZER, M. Provenance-Aware Storage Systems. In *USENIX* (May 2006).

[12] ROUSSEL, N., TABARD, A., AND LETONDAL, C. All you need is log. In *Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection* (May 2006).

[13] SHAH, S., SOULES, C., GANGER, G., AND NOBLE, B. Using Provenance to Aid in Personal File Search. In *USENIX* (2007).