Zooming Metadata: A Framework for Facilitating Provenance Exploration

Linus Karsai
University of Sydney

***Abstract*—I describe a framework that can be used to improve the exploration of complex and large provenance graphs. This framework identifies tasks and features that should be implemented in order to facilitate the exploration of provenance information. I used a prototype that implements a subset of these features in order to partially evaluate the framework.**

I. Introduction

Provenance is the lineage of an entity: a list of what influenced and effected an object before it got to the stage it's currently in. The concept of provenance isn't new[3] and is historically referenced in accounting or art where it's used to describe the record of ownership of an antique.

In the context of digital information though it's quite a new field of research[13] and describes the form of lineage associated with a digital artefact. By viewing the provenance of a file you can identify and explore what other pieces of data have influenced the file and what chain of events led to the file's current state.

For example: say you're using a script in order to create visualisations for a report. And the script has a series of parameters that can be altered in order to alter the visualisation's appearance. You come back to rewrite your report a couple of months later because your original data was erroneous and you're trying to recreate the visualisation, however you can't remember the parameters you originally used. Instead of working out the parameters through trial and error, the provenance of the original visualisation could be used to show the state of the script that originally created it as well as what parameters where used.

Research in the area of digital provenance is split into five primary fields:

A. Inception

Because research into digital provenance is still in it's infancy, there's a series of inception papers that explore the concept of digital provenance[7, 9]. This research goes into great detail concerning the different concepts related to provenance as well as providing some detailed examples of how provenance can be used in the real world. This is a great place to start in order to research provenance and its potential.

Please note that as part of my understanding of the assignment specifications, a large portion of this paper is fictional.

B. Acquisition

As mentioned before, provenance is the lineage of a file: everything that effected the file into its current state. This can be recorded at various different levels, and each level has different approaches to recording with different levels of verbosity. For example provenance can be recorded at OS level[14] in which case each process and file is a provenance artefact or it can also be recorded at application level[10, 12, 2] in which case different segments of an application attribute more relevant provenance however it often requires more effort from developers in comparison to OS level capturing.

C. Storage

A lot of papers reference provenance been stored locally, usually in a plaintext format[4]. There's also research into how to implement provenance into cloud and distributed solutions[15]. This allows the provenance associated with database files to also be stored in the same medium.

D. Security

Security is a complex issue in provenance because a lot of information can be implied from relationships between different nodes. So even if a nodes information is censored, the relationship that node has with other nodes sill leaks information. Different access controls are required in order to limit who has access to different relationships in a provenance graph[6, 8]. There's also research into systems like block chains in order to prevent history modification[11].

E. Display

Provenance graphs are most often displayed as directional acyclic graphs (DAGS). Visualisation is difficult problem because the graphs quickly become large and unwieldy. Related research has been done in visualising generic large graphs[1, 16], but there's also specific research done in simplifying provenance visualisation either through clustering, zooming[17] or user-defined views[5].

II. Problem

How to display provenance to users in a way that's understandable? Eluded to before, provenance never reduces in size. This means that it is forever expanding and been added too. Quickly issues arrise when trying to present such a large amount of information to users, the graphs can become so large and unweildly that it's impossible to identify usefull patterns or connections. You can see in Figure 1 an example of how crowded graphs can become, and this is still a small amount of provenance, the number of nodes can strech into thousands and millions.

An interface for visualising provenance information would have to overcome the following technical and conceptual hurdles:

A. Zomming and Clustering

Provenance graphs are simply too large. Some sort of zooming/clustering algorithm is required in order to group relevant nodes and allow the user to conceptually model the information. The issue then is to logically label a cluster so that users can understand its content.

B. Security Obfuscation

Previous research into provenance security shows that previous access control methods won't fully secure provenance graphs, primarily the issue of leaked data from shown relationships. This suggests that access controls will be expanded to include relationships and even subgraphs. From a visualisation perspective this means that it won't be uncommon to present partiallly obscured graphs to users. How to present those graphs and inform users that they're viwing possibly summarised information is a problem not yet apporached by exsising visualisers.

C. Ease of use

Provenance information is primarily a form of metadata. This means that like most metadata it will be used intermitantly. This means the application must be intuative enough that infrequent use doesn't have a neggative effect uppon productivity.

III. Solution

My solution is a framework that allows easy exploration and examination of provenance graphs. Below I describe each task required by the implementation in general form as well as possible implementations. Proposing a framework creates ground for conversation for provenance visualisation as well as identifying a blueprint that can be modified and updated in order to accommodate for future unpredicted use cases. The tasks are split into three areas: Exploration, Annotation and Security.

A. Exploration

It's vital that users are able to explore provenance data as well as find specific information. A lot of the following tasks are similar to those suggested by Ben Shneiderman for information visualisation so I recommend reading his paper "The Eyes Have It"[18] for more insight.

1) Overview: Gain an overview of the entire graph. Whilst this seems counter-intuitive at first, it's important that users are able to see the entire graph at once in order to create a mental model of the information. The most common implementation of this is a simple slider that controls zoom factor and a window that allows panning around the graph. In order to view large graphs a clustering technique can be used to group similar nodes.

2) Zoom/Details on Demand: Users should be able to zoom in on clusters and nodes, viewing relevant information about them. In the case where nodes are clustered together, double clicking can be used to expand and view sub-nodes. A sidebar is an easy way of showing details about a node.

3) Peripheral: Users should have an indication of where they are relative to the rest of the graph so they can identify context. Two common methods of doing this are through a thumbnail in the corner of the interface that shows the entire graph with a box representing your viewport, or a fisheye interface that shows everything on screen with nodes furthest from your point of interest appearing smallest.

4) Filter: In order to find specific information it's important that users have a way of filtering information relevant to them. This can either be implemented through a search bar or more thoroughly through a series of filter controls for different aspects: time, node type etc.

5) History: Users should be able to see the steps they've taken to get to their current view. This allows for quickly undoing exploration steps. A good example of this can be seen in the Adobe Photoshop history page (Figure 2).

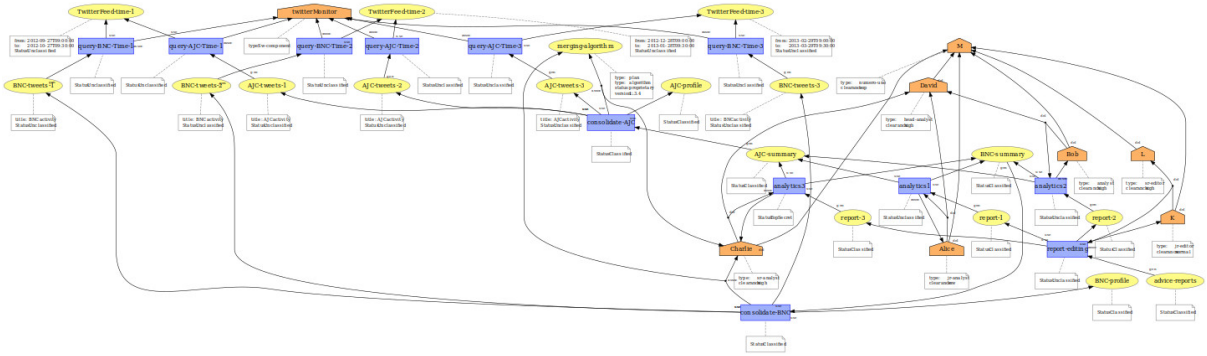

Fig. 1. Complex provenance example.

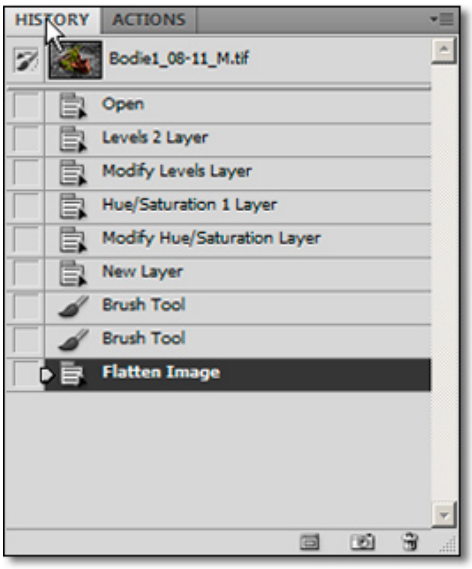

Fig. 2. Adobe Photoshop History Pannel

B. Annotation

Annotation is important as it allows users to quickly catch up on previous work they have previously committed to a provenance graph. This means that the time required to get up to speed with a graph after extended periods is limited because users can read up on previous annotations.

1) Points of Interest: Users should be able to mark points that they find interesting allowing further exploration later. This can be implemented through a bookmark panel that allows points of interest to be marked with custom titles/tags.

2) General Annotation: General annotation allows users to describe or take notes on certain parts of the graph. In it's most simple form annotations can be text snippets attached to relevant nodes or groups of nodes. They should be exportable.

3) Exporting and sharing: Provenance at the moment has limited traction outside of it's relevant research communities. It's therefore important that provenance graphs or sub graphs can be exported so that they can be shared and analysed by other people. This can be implemented by allowing exportation in multiple open source formats: PROV Model or CSV.

C. Security

A large portion of provenance information can be sensitive and it's important to take this into account when creating an interface.

1) Access control: A view for administrators to be able to view who has access to different parts of a provenance graph. This information can be presented as an overlay on the standard graph.

2) Partial graphs: Sometimes entire sub graphs will be out of a users access level, the interface needs to be able to present graphs that have sections missing. Feedback must be given to the user that they have information hidden from them.

3) Security feedback: Users should have feedback on whether the graph could be misleading because of hidden information. As well as a way of indicating which provenance sources are most reliable (for example, human input is less reliable than automatic digital sources). This can be done with tool tips and warning banners.

IV. Results, Discussion and Analysis

To support the above framework I created a prototype interface that implements a subset of the features. First I'll talk about how the prototype was implemented, then the methods I used to get qualitative/quantitative feedback as well as analysis of those results.

A. Prototype

Playing to my strengths of web-development I created a web prototype using D3 a javascript module for visualisation. Using a web interface also means that my prototype isn't limited to a certain operating system or end use device. Following is the list and description of the features my interface implements.

- **Overview:** A thumbnail in the bottom right corner with a box representing your view port.
- **Clustering:** Groups of relevant nodes are clustered either manually or via a time heuristic.
- **Filtering:** Users can search for text in nodes as well as filter down to certain time spans.
- **History:** A sidebar shows a list of past exploration actions.
- **Points of interest:** Users can bookmark nodes through a sidebar.

B. Usability Evaluation Study

A series of tasks where designed in order to measure how effective users found different features of the prototype. Users where asked to complete the tasks once in a think aloud environment, then return two weeks later to complete the same tasks in order to measure how useful the interface is with infrequent use.

The think aloud took on average an hour to complete all tasks. Each task was timed and if users went over a limit for each question then they where told they could skip it in order to not bias later results through frustration.

After the think aloud users filled in a quick survey regarding their personal reactions to the interface, as well as a general questionnaire identifying demographics and proficiency with computers.

C. Analysis

Seven of the eight users I interviewed had never head of provenance before and none of them knew of the concept of provenance relating to digital data.

Of the tasks that required users to identify the main contributors of a certain file 75% of participants where able to accomplish the task in under 3 minutes. Some participants noted they found if difficult because they had never used graphs before.

> "I'm not quite sure what all these circles are."

However it seems that once users had gotten over this initial hurdle, they then had no issues identifying specific nodes in the graph. When they returned two weeks later all users where able to find the relevant information in under 3 minutes.

People who use image manipulation applications such as Adobe Photoshop seemed to find it easier to move around the graph as my prototype shares many of the same design choices such as a history pane and thumbnail overview.

The history bar was one of the most useful features with users interacting with it in a majority of the tasks (6 users interacted with the history sidebar for over 90% of tasks).

Users didn't use bookmarks to pinpoint locations unless it was specified in the task that they should mark a point so that they could return to it at a later date. Qualitative feedback from experts suggested that the bookmark feature was crucial, so this unexpected result could be attributed to users not having to work with the same provenance graph for extended periods.

V. Conclusion and Future Work

In this paper I outlined a set of features recommended for provenance exploration. The focus of these features is on usability, interaction with others and future security concerns. I intend for these to open up a discussion in the field about features required by people who analyse provenance everyday as well as provide a starting point for people creating interfaces.

Over time it is likely that this list of features will need to be tweaked and altered as a more homogeneous group of people use provenance. I hope this framework will be an evolving document that is adapted as new possibilities for provenance are discovered.

Initial feedback from the prototype was positive. Future work would be to extend this prototype to include all the features outlined in my framework.

REFERENCES

[1] James Abello, Frank Van Ham, and Neeraj Krishnan. ASK-GraphView: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006.

[2] M David Allen, Adriane Chapman, Barbara Blaustein, and Len Seligman. Capturing Provenance in the Wild 1 The Challenge of " Open World " Provenance Capture. *Management*, pages 98–101, 2010.

[3] David Bearman and Richard Lytle. The Power of the Principle of Provenance. *Archivaria*, 21(February 1982):14–27, 1985.

[4] Khalid Belhajjame, Helena Deus, Daniel Garijo, Graham Klyne, Paolo Missier, Stian Soiland-Reyes, and Stephan Zednik. PROV Model Primer, 2013.

[5] O Biton, S Cohen Boulakia, and S B Davidson. Zoom*UserViews: Querying Relevant Provenance in Workflow Systems. *Vldb*, pages 1366–1369, 2007.

[6] Uri Braun, Avraham Shinnar, and Margo Seltzer. Securing Provenance. *Proceedings of the 3rd conference on Hot topics in security (HOTSEC'08)*, pages 1–5, 2008.

[7] Lucian Carata, Sherif Akoush, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Margo Seltzer, and Andy Hopper. A Primer on Provenance. *Queue*, 12(3):10–23, 2014.

[8] James Cheney. A formal framework for provenance security. *Proceedings - IEEE Computer Security Foundations Symposium*, pages 281–293, 2011.

[9] Paul Groth, Yolanda Gil, James Cheney, and Simon Miles. Requirements for Provenance on the Web. 7(1):39–56, 2012.

[10] Pj Guo and M Seltzer. BURRITO : Wrapping Your Lab Notebook in Computational Infrastructure. *Proceedings of the 4th USENIX conference on ...*, page 4, 2012.

[11] Ragib Hasan, Radu Sion, and Marianne Winslett. Preventing history forgery with secure provenance. *ACM Transactions on Storage*, 5(4):1–43, 2009.

[12] Peter Macko and M Seltzer. A general-purpose provenance library. *... on the Theory and Practice of Provenance*, 2012.

[13] Luc Moreau. The Foundations for Provenance on the Web. 2, 2010.

[14] Kiran-Kumar Muniswamy-Reddy, David A Holland, Uri Braun, and Margo Seltzer. Provenance-aware Storage Systems. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*, ATEC '06, page 4, Berkeley, CA, USA, 2006. USENIX Association.

[15] Kiran-Kumar Muniswamy-Reddy, Peter Macko, and Margo Seltzer. Provenance for the Cloud. *Proceedings of the 8th USENIX Conference on File and Storage Technologies*, pages 14–15, 2010.

[16] Doug Schaffer, Zhengping Zuo, Saul Greenberg, Lyn Bartram, John Dill, Shelli Dubs, and Mark Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human Interaction*, 3(2):162–188, 1996.

[17] M.I. Seltzer and Peter Macko. Provenance Map Orbiter: Interactive Exploration of Large Provenance Graphs. *Proceedings of the 3rd USENIX Workshop on the Theory and Practice of Provenance (TaPP'11), June*, pages 20–21, 2011.

[18] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. *Visual Languages, 1996. Proceedings., IEEE ...*, pages 336–343, 1996.