It's About the Data: Provenance as a Tool for Assessing Data Fitness

Adriane Chapman, M. David Allen, Barbara Blaustein
{achapman, dmallen, bblaustein}@mitre.org
The MITRE Corporation

Abstract

The end goal of provenance is to assist users in understanding their data: How was it created? When? By whom? How was it manipulated? In other words, provenance is a powerful tool to help users answer the question, "Is this data fit for use?" However, there is no one set of criteria that make data "fit for use". The criteria depend on the user, the task at hand, and the current situation. In this work we describe Fitness Widgets, predefined queries over provenance graphs that users can customize to determine data fitness. We have implemented Fitness Widgets in our provenance system, PLUS.

1. Introduction

Those who work with provenance sometimes forget that provenance is not a goal or need in and of itself, but a technical approach employed to satisfy data needs and goals. The ultimate, *data-centric*, goal is to build tools and applications over provenance information to support a user's needs. One of the major classes of user questions where provenance can help is in evaluating whether data is fit for a specific purpose; e.g., does the data item derive from an Internet source? Were untrusted organizations involved in producing the data item? There are many problems with data that derive from the way the data was produced in the first place, and provenance is well-suited to ferreting those problems out.

Provenance "in the raw" is not always useful to users; it is generally presented as a directed acyclic graph (DAG). While many users have a good intuitive understanding of simple graphs, provenance graphs are often large and unwieldy. Efforts such as [6, 11] make the graph more user friendly by abstracting away repetitive or "uninteresting" bits. Additionally, there are provenance query languages [4, 8] and graph query languages [1, 7] through which expressive queries over the provenance information can be expressed. These are good first steps, however we posit that for most data-centric tasks and most users, viewing or interacting with a complex provenance graph, or a query language, is itself a non-starter. Users benefit from provenance because of computations done on top of that provenance, not from the graphs themselves, just as all users benefit from data structures like internet routing tables, while few ever see them. The value of provenance is that it enables certain novel kinds of analytics; the value is not the raw provenance data itself.

What is really needed is a system that can assess the fitness of an item of data, but there is no single set of criteria for data fitness (sometimes also framed as "trustworthiness"). Fitness criteria depend on the user and the application, as well as on organizational policies and on the nature and urgency of the user's task. For tasks with serious consequences, a user is more likely to need to verify that authoritative data sources were used in the final product. For a "quick snapshot" task needing general assessments of current events, the user may be more concerned about the timeliness of the data, even if it is not exhaustively vetted.

The provenance graph in Figure 1 follows a simple intelligence analysis example, showing how data from different sources is combined to produce analysis products. Ann and Bob have access to two reports, shown in the graph as Analysis Product and Revised Analysis Product. Which is appropriate for their use? Ann is concerned about foreign government misdirection. She does not wish to rely upon any data that was owned or held by a foreign source. Bob on the other hand, wishes to use the most current resource. In this scenario, Ann should choose the Analysis Product and Bob the Revised Analysis Product.

In this work, we discuss *Fitness Widgets*, a method for users to get answers to data-centric questions over provenance to determine if a data item is fit for a specific use, without needing to navigate the provenance graph. Provenance graphs then become auxiliary databases, whose usefulness derives from their ability to answer fitness questions. Fitness Widgets are small software modules that can be customized by users to reflect the requirements of the task at hand.

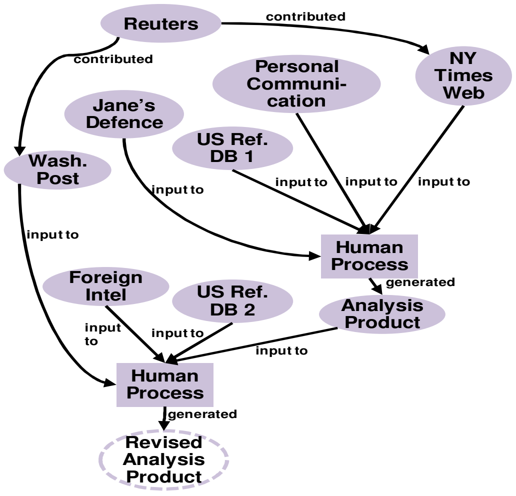


Figure 1: The provenance graph for a revised analysis product. Public sources (Reuters, Washington Post, New York Times) all contributed to the final product. Authoritative sources (US Reference Database 1 and 2, those blessed by a particular organization), are also used. There are also personal communications and foreign intelligence sources. Rectangles represent processes; ovals data.

2. Foundations

A provenance graph is a directed acyclic graph (DAG), $G = (N, E)$, containing a set of nodes, N, and a set of edges, E. Each node has a set of features describing the process or data it represents, e.g., timestamp, description, etc. Edges in the graph denote relationships, such as usedBy, generated, inputTo, etc., between nodes as influenced by OPM [12].

Provenance nodes representing data can refer to any sort of object, for example files, XML messages, relational data items of arbitrary granularity (table, row, column, cell), etc. The data itself is not stored in the provenance system for security and archiving reasons. However, the provenance may contain additional "breadcrumbs," such as access method and identifier, to allow users to access the underlying information. Examples of these access methods or identifiers might include a URL, or SQL. Users may also annotate anything in a provenance graph with additional metadata.

A provenance node maintains a core set of attributes, including a unique id, timestamp and description. However, the provenance node can be extended to contain any number of additional attributes, such as owner, user, etc. The extensions required depend on the application needs of the data and system that the provenance system is supporting. For instance, battery charge level may be a necessary provenance extension in a use case that uses provenance to determine the accuracy of sensor readings.

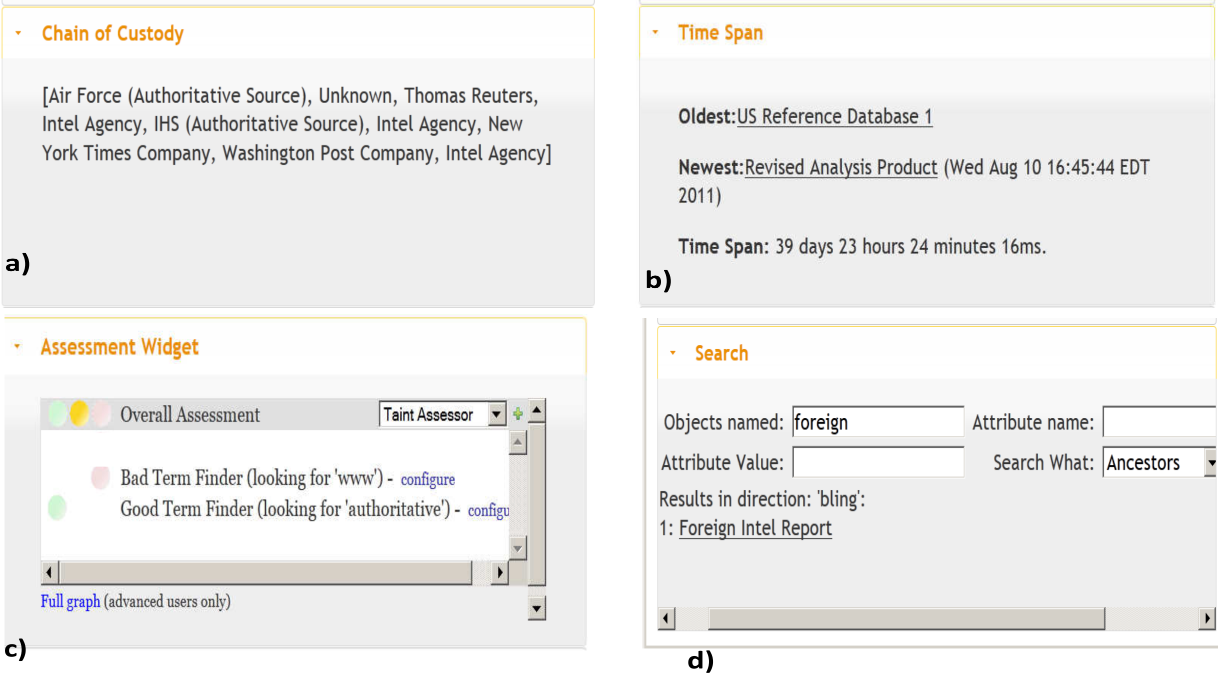
As a result of this model, Fitness Widgets have at their disposal node and edge information which permits relationship tracing, and attributes containing user-specified information specific to the application domain.

3. Fitness Widgets

A Fitness Widget is a query, Q, that operates over a provenance graph G, that looks for a set of given conditions using information in G. There is a trade-off between the power of a Fitness Widget to perform some complex analysis, and the ease of a user specifying it in the first place. While it is possible to write a complex Fitness Widget that encapsulates an organization's entire data policy, doing that would require a developer to write custom code. As stated earlier, because fitness for use is dependent upon the user, need, situation and task at hand, it is imperative that users are given a simple means to express their own Fitness Widgets based on their current situations. As such, two distinct types of Fitness Widgets are important to consider: complex predefined, and user-defined on the fly.

Complex predefined Fitness Widgets based on customer interest are necessary to encapsulate more complex business logic; two examples are shown in Figure 2a-b. Figure 2a shows a Fitness Widget describing the Chain of Custody for a given item, while Figure 2b shows a Fitness Widget calculating the time span for a data item based on the age of its ancestors. The queries needed to generate this information are more advanced than a typical non-CS user could write, and would generally be developed for a group of users.

The key feature of Fitness Widgets is their focus on ease of creation and application by everyday users, illustrated in Figure 2c-d. The term Fitness Widget in Figure 2c looks for specific user defined triggers. Notice that the Fitness Widget indicates that both the bad term (red stop light) and the good term (green stop light) are present. The customizable Fitness Widgets include a set of parameterized queries over the provenance. In our current prototype, these include a taint assessor, good-term and bad-term finders, and a twitter finder. The taint assessor checks all ancestors of a data item and shows a red light if any data item in the lineage was marked tainted (see [3] for a discussion of taint propagation). The good term and bad term finders will search for a specific term and display green or red lights respectively if that term is found. When the Fitness Widget is run, the user is shown a series of subqueries and their results, as in Figure 2c. Because a user may choose to value different subqueries



[Air Force (Authoritative Source), Unknown, Thomas Reuters, Intel Agency, IHS (Authoritative Source), Intel Agency, New York Times Company, Washington Post Company, Intel Agency]

Oldest: US Reference Database 1

Newest: Revised Analysis Product (Wed Aug 10 16:45:44 EDT 2011)

Time Span: 39 days 23 hours 24 minutes 16ms.

Overall Assessment — Taint Assessor

Bad Term Finder (looking for 'www') - configure

Good Term Finder (looking for 'authoritative') - configu

Full graph (advanced users only)

Objects named: foreign
Attribute name:
Attribute Value:
Search What: Ancestors
Results in direction: 'bling':
1: Foreign Intel Report

Figure 2: Samples of Fitness Widgets. a) The pre-defined "Chain of Custody" lists the owners of all data and processes shown among the ancestors of a particular node – in this case, the "Revised Analysis Product" in Figure 1. b) The pre-defined "Time Span shows the difference in timestamps among nodes in a specific path. c) The configurable "Assessment Widget" looks for specified desirable and undesirable terms among ancestors of a node. d) "Search" allows users to specify more detail.

more highly than others, we use a simple, *customizable*, policy to combine these parameterized query results into the "Overall Assessment". (The default is that if all queries show green, so does the overall assessment; if there are mixed red and green, the overall assessment shows yellow, etc.) It is up to users to synthesize these results, along with those for other queries such as Chain of Custody and Time Span, into meaningful decisions based on their data needs and current context. If a user believes that a single customized policy to combine results will be generally applicable, it is, of course, possible to add this policy into the user's Fitness Widget.

Fitness Widgets are a simple means to assess different data items automatically against the same criteria. Figure 2d shows an example of a Search Term Fitness Widget applied to the Revised Analysis Product. It succinctly shows that it includes a foreign source.

So far, we have discussed Fitness Widgets as an assessment tool that a user invokes after choosing a data item of interest. However, Fitness Widgets are also well suited to data discovery. In our earlier example, suppose that Ann started by considering the Revised Analysis Product. She used a customized Fitness Widget reflecting her criterion that no foreign-supplied data be used. She could then use the Fitness Widget search capability to search for other analysis products, finding the one that meets her criteria. With Fitness Widget searches, a user can specify not only the type of data of interest, e.g. by keyword or other data attributes, but also which Fitness Widgets must be satisfied for any data item returned.

4. Implementation

We have implemented Fitness Widgets within the PLUS system [5], a provenance manager developed at The MITRE Corporation to address the previously unmet requirements shared by most of our U.S. government customers.

Once provenance information is captured, it must be stored for later use. PLUS can be run as a stand-alone manager with a centralized repository or as a set of provenance managers and repositories distributed across organizations [2]. PLUS uses a MySQL database for provenance storage, and it models provenance similar to the emerging W3C Provenance standard [14]. It is assumed that the capture mechanisms have been tuned to the expected use cases for a given group or organization.

5. Related Work

Current tools that exist for making provenance more usable can be classified into two categories: viewers and subsequent applications. Viewers take a large amount of provenance information, and present a graph containing only a subset of it back to the user [6, 11, 13]. These viewers assume that the user wishes to interact with a graph, and work to make that graph more manageable. An alternative approach is via subsequent applications. These applications recognize that provenance is often too complex for a user to view and understand, but wish to assist the user in performing advanced functions. For instance, [3] uses the provenance information, without showing it to the end user, to warn of possible suspicious users and behaviors. Meanwhile, [9] uses provenance to enable executable papers; the users themselves do not need to inspect provenance to reproduce results, it is done automatically. Finally, [10] recommends possible alternative execution strategies for visualizing scientific data, based on knowledge gained through the provenance of previous visualization attempts.

Of relevance to this work is the work on graph query languages [1, 7] and provenance query languages [4, 8]. These works provide a semantically precise way to express queries over graph and provenance information respectively. Fitness Widgets, which are pre-defined queries over provenance information, are a natural assistance to non-CS users creating queries using these technologies.

6. Future Work and Conclusions

Fitness Widgets are predefined queries that are applied to a provenance graph. There are three main areas for expansion of this work: query expressibility, or integration with a provenance query language such as ProQL [8], user assistance and incorporating information from the underlying data.

Currently, Fitness Widgets are small software modules which wrap analytical queries across provenance graphs. This idea could be extended to transform them into proper graph queries, so that anything expressible in that graph query language can be written as a Fitness Widget.

Finally, Fitness Widgets currently only operate over the provenance data. While the original data is not stored with the provenance, or security and archiving reasons, “breadcrumbs” back to the original data exist. A hybrid system that integrates provenance with the historical data, and permits Fitness Widgets on the basis of mixed provenance and data information would expand the ability of the user to judge fitness for use.

In this work, we use Fitness Widgets to expose a fundamentally new kind of data source: provenance. Because the end goal of provenance should be an improved *data-centric* view, they allow the user to express desirable and undesirable data properties. Fitness Widgets must be customizable by the end user to reflect the fitness criteria for specific users, situations, and tasks.

7. Bibliography

[1] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener, "The Lorel query language for semistructured data," *IJDL*, vol. 1, 1997.

[2] M. D. Allen, A. Chapman, B. Blaustein, and L. Seligman, "Getting It Together: Enabling Multi-organization Provenance Exchange," *TaPP*, 2011.

[3] M. D. Allen, A. Chapman, L. Seligman, and B. Blaustein, "Provenance for Collaboration: Detecting Suspicious Behaviors and Assessing Trust in Information," *CollabCom*, 2011.

[4] M. K. Anand, S. Bowers, and B. Ludascher, "Provenance Browser: Displaying and Querying Scientific Workflow Provenance Graphs," *ICDE*, 2010.

[5] A. Chapman, M. D. Allen, B. Blaustein, and L. Seligman, "PLUS: A Provenance Manager for Integrated Information," *IEEE International Conference on Information Reuse and Integration (IRI ‘11)*, 2011.

[6] S. Cohen-Boulakia, O. Biton, S. Cohen, and S. Davidson, "Addressing the provenance challenge using ZOOM," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 497-506, 2008.

[7] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl, "RQL: a declarative query language for RDF," *WWW*, 2002.

[8] G. Karvounarakis, Z. G. Ives, and V. Tannen, "Querying Data Provenance," *SIGMOD*, 2010.

[9] D. Koop, E. Santos, P. Mates, H. T. Vo, P. Bonnet, B. Bauer, B. Surer, M. Troyer, D. N. Williams, J. E. Tohline, J. Freire, and C. T. Silva, "A provenance-based infrastructure for creating executable papers," *ICCS*, pp. 648–657, 2011.

[10] D. Koop, C. E. Scheidegger, S. P. Callahan, J. Freire, and C. T. Silva, "VisComplete: Automating suggestions for visualization pipelines," *IEEE Trans. Vis. Comp. Graph.*, vol. 14, pp. 1691–1698, 2008.

[11] P. Macko and M. Seltzer, "Provenance Map Orbiter: Interactive Exploration of Large Provenance Graphs," *TaPP*, 2011.

[12] L. Moreau, J. Freire, J. Futrelle, R. McGrath, J. Myers, and P. Paulson, "The Open Provenance Model," University of Southampton 2007.

[13] N. D. Rio, P. P. d. Silva, and H. Porras, "Browsing Proof Markup Language Provenance: Enhancing the Experience," *IPAW*, pp. 274-276, 2010.

[14] W3CProvenance, "http://www.w3.org/2011/prov/wiki/Main_Page," 2012.