The Providence of Provenance

Peter Buneman

School of Informatics,
University of Edinburgh

Abstract. For many years and under various names, provenance has been modelled, theorised about, standardised and implemented in various ways; it has become part of mainstream database research. Moreover, the topic has now infected nearly every branch of computer science: provenance is a problem for everyone. But what exactly is the problem? And has the copious research had any real effect on how we use databases or, more generally, how we use computers.

This is a brief attempt to summarise the research on provenance and what practical impact it has had. Although much of the research has yet to come to market, there is an increasing interest in the topic from industry; moreover, it has had a surprising impact in tangential areas such as data integration and data citation. However, we are still lacking basic tools to deal with provenance and we need a culture shift if ever we are to make full use of the technology that has already been developed.

1 Why Were We Interested?

It is well over 20 years since the issue of provenance was first introduced to, and initially ignored by, the database community [28]. It is noteworthy that much of the early work on this topic [28,12,9] was initiated by the need to understand provenance in the context of data integration. In fact, the earliest [28] paper states this very well:

> although the users want the simplicity of making a query as if it were a single large database, they also want the ability to know the source of each piece of data retrieved.

The other early paper on the topic [29] was more concerned with scientific programming, but even this addressed the provenance of "each piece of data" in an array. My own stimulus for studying provenance came from some molecular biologists with whom we were collaborating on data integration. The problem was compounded by the fact that they were not just building a data warehouse but also manually correcting and augmenting the curated database they were publishing.

Although database queries are relatively simple programs – they do little more than rearrange the data from their input – the "source of each piece of data retrieved" turns out to be a rather elusive concept. At least there are numerous ways of describing the source: one may ask *why* a tuple is in the output [12,9] or

G. Gottlob et al. (Eds.): BNCOD 2013, LNCS 7968, pp. 7–12, 2013.

how it was constructed [19,21]. Even the description of *where* a "piece of data" has been copied from is non-trivial [7].

At the same time that the initial research on provenance and database queries was taking off, computer science researchers involved in scientific workflows started to think about provenance, and now several establish workflow systems provide some form of provenance capture [30,5,4,13,4,22]. Anyone who has done any large-scale, complex, scientific programming, such as data analysis or simulation will know how easy it is to lose track of one's efforts. One wants to be able to repeat the work, and this calls for some form of provenance. Here we are in a different playing-field. First, we are interested in the whole output, not one "piece of data"; second, the subroutines that one invokes may be much more complicated than database queries; and, finally, one may not trust those subroutines. To elaborate on this last point, database query languages have precise specifications, and we expect them to perform according to those specifications so we may not need to record the details of their execution. This is not the case with scientific programs in general.

To deal with this, and perhaps to deal with other concepts in provenance – such as those arising from traditional provenance associated with historical artifacts – a W3C working group has produced a series of models of provenance [23,17]. Starting with a simple causal model of provenance, the model has been enriched with a variety of notions that describe various forms of interaction between processes and artifacts. For whatever reason, there has been little attempt to connect these models of provenance, which sometimes go under the name of "workflow provenance" with the languages e.g., the workflow languages, that produce provenance graphs. And while there are some obvious and basic connections between workflow and data provenance, there may be much more to be done in this area.

2 What Have We Achieved?

Possibly the greatest achievement has been the drawing together of a number of research communities to tackle what, at first sight, is a common problem. At a recent provenance workshop [1] there were people from the semantic Web, databases, programming languages, computer systems, scientific programming as well as curators who think about provenance in its traditional sense. In addition, the study of provenance has led to some solid technical connections. Program slicing, which attempts – for the purposes of debugging – to provide a part of trace of the execution of a program that is relevant to some part of the output, is closely connected with provenance and database queries [11]. In a series of papers [19,21,1] Tannen and his colleagues show how many of the extensions to relational databases: c-tables, probabilistic databases, multiset semantics, as well as a number of provenance models have a common and powerful generalization as an abstract "provenance polynomial" associated with each tuple in the answer

[1] TaPP'11. 3rd Usenix Workshop on the Theory and Practice of Provenance. Heraklion, June 2011.

to a query. Moreover, formalisms developed for provenance have influenced other areas such as data integration [18] and hypothetical query answering [15].

The W3C working groups have also achieved some form of consensus on a general model of provenance [23,17]. One of the things that is discussed in this work but has not yet been fully formalised is the notion of *granularity*. Provenance graphs should be capable of expansion into an increasingly fine-grained description and possibly connect with ideas of data provenance. This does not appear to be difficult, but the details do need to be worked out.

This process of repeated expansion of provenance raises the important question: How much provenance we should record? Why not record the workflow specification or the source code together with the input and re-execute the process if we want to know what happened? If we look at data provenance where the programs are queries, the details of "what happened" are typically not of interest. Database queries are declarative, and we assume they have performed according to the well-defined specification. As we have already noted, not all programs have a well-defined specification and those that do may contain errors. And, of course, if the purpose of keeping provenance is to "instrument" the execution of a program in order to understand performance [6,24], then the more detail the better.

In practice, we generally know what parts of a workflow are error-prone or are the cause of a performance bottleneck, so why not keep provenance for just those components, and ignore it for those that are properly specified? In doing the latter, there is, of course, an interesting interaction with privacy in workflows [14].

3 What Impact Have We Had?

Those of us who have been working on provenance for some time might pose the alternative question with: What impact did we expect it to have? It would be great if we could say that a major disaster was avoided because we kept provenance, and we probably can say this. The excellent version control systems that are used in software development already keep a form of provenance, and their use has almost certainly avoided a major software disaster, but we don't find this remarkable. In a related lecture, Margo Selzer [26] rates the impact of provenance research as zero – at least in the sense that it has not yet produced any headline-grabbing results.

Are we expecting too much? From my own experience, the recording of provenance information has sometimes made life easier for scientists but has seldom enabled them to do things that they could not, by more laborious means, already do. Here are two examples.

- In working with the IUPHAR database [27] we installed an archiving system [8] that recorded the complete history of the database. Proper archiving is an essential precursor of any provenance or citation system that involves an evolving database. One of the properties of the system, which was intended as a compression technique, is that it allows one to see the evolution of some *part* of the database. This was useful to the curators who did not

have to open up, and search, old versions of the database in order to see how some table had evolved.
- There was a recent case [2] of an influential paper, which justifies the case for economic austerity measures being found by a student, who had been asked to reproduce the results, to contain significant errors. It required a lot of work by the student, who ultimately had to consult the authors, to do this. One could imagine that had the paper been "provenance enabled", or – better – "executable" [25,16], the student would have spotted the mistake immediately. Indeed the mistake might never have happened!

So perhaps the real impact of provenance is not so much in the tools and models that we are directly developing to deal with it, but in the ancillary work on archiving, data citation, annotation, executable papers, program slicing, etc. that are directly connected with it and that are being informed, to some extent, by those models and tools.

4 A Change of Attitude Is Needed

Although the scientific programming community is, of necessity, becoming aware of the need to record provenance in some form. It is unclear that other communities, such as the Semantic Web, have developed the same level of awareness. Moreover, most of us are blissfully unaware that we are constantly throwing away provenance information in our ordinary work. To take one example the "LOD cloud" [20] – a distributed collection of billions of RDF triples – was created mostly by transforming and linking existing data sets. Somewhere in this linkage you will find the data extracted from the CIA World Factbook [3], which gives the population of Afghanistan as 31,889,923. If you go the the CIA World Factbook [10], you will find the population of Afghanistan as 30,419.928, with an annotation indicating that this estimate was made in 2012 and a further annotation indicating that this is significantly different from a previous estimate of 33,809,937[2] Presumably the LOD figure was copied from an older version of the Factbook, but the version (provenance) is not obviously recorded and, more importantly, whatever annotations there were in the original have "fallen off" in the process that created the relevant LOD triples.

It is a non-trivial challenge to make this process of transforming and linking data "provenance aware", it appeard to require a substantial modification or even a complete restructuring of the RDF. The same situation appears in any kind of data warehousing operation, but until we tackle it properly, we have to assume that almost any data we find on the Web is stale.

At a higher level, we need to get provenance into the general consciousness. The evils of copy-paste need not be repeated here; but imagine a situation in which, whenever you did a copy operation, you automatically were given provenance data, and whenever you did a paste, either the program you were using (e.g. a text editor) knew what to do with the information or you had consciously

[2] Updated to a 2013 estimate of 31,108,077.

to throw the provenance information away. Even in writing this very short paper, I performed tens of copy-paste operations, and even here the benefits of keeping provenance data are self-evident; but the process of keeping provenance (in footnotes and citations for example) is arbitrary, and the editors and text formatters that I used gave little help with the process.

So perhaps the practical moral of these stories is that we should worry less about what provenance is and concentrate more on what we can do with it once we have it.

References

1. Amsterdamer, Y., Deutch, D., Tannen, V.: Provenance for aggregate queries. CoRR, abs/1101.1110 (2011)
2. http://www.bbc.co.uk/news/magazine-22223190
3. Bizer, C.: World factbook, fu berlin (UTC) (retrieved 16:30, May 4, 2013)
4. Bowers, S., McPhillips, T.M., Ludäscher, B.: Provenance in collection-oriented scientific workflows. Concurrency and Computation: Practice and Experience 20(5), 519–529 (2008)
5. Bowers, S., McPhillips, T., Ludäscher, B., Cohen, S., Davidson, S.B.: A model for user-oriented data provenance in pipelined scientific workflows. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 133–147. Springer, Heidelberg (2006)
6. Braun, U., Shinnar, A., Seltzer, M.I.: Securing provenance. In: HotSec (2008)
7. Buneman, P., Cheney, J., Vansummeren, S.: On the expressiveness of implicit provenance in query and update languages. ACM Trans. Database Syst. 33(4) (2008)
8. Buneman, P., Khanna, S., Tajima, K., Tan, W.C.: Archiving scientific data. ACM Trans. Database Syst. 29, 2–42 (2004)
9. Buneman, P., Khanna, S., Tan, W.-C.: Why and where: A characterization of data provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 316–330. Springer, Heidelberg (2000)
10. Central Intelligence Agency. The World Factbook, https://www.cia.gov/library/publications/the-world-factbook/
11. Cheney, J., Ahmed, A., Acar, U.A.: Provenance as dependency analysis. Mathematical Structures in Computer Science 21(6), 1301–1337 (2011)
12. Cui, Y., Widom, J.: Practical lineage tracing in data warehouses. In: ICDE, pp. 367–378 (2000)
13. Davidson, S.B., Freire, J.: Provenance and scientific workflows: challenges and opportunities. In: SIGMOD Conference, pp. 1345–1350 (2008)
14. Davidson, S.B., Khanna, S., Roy, S., Stoyanovich, J., Tannen, V., Chen, Y.: On provenance and privacy. In: ICDT, pp. 3–10 (2011)
15. Deutch, D., Ives, Z., Milo, T., Tannen, V.: Caravan: Provisioning for what-if analysis. In: CIDR (2013)
16. Freire, J., Silva, C.T.: Making computations and publications reproducible with vistrails. Computing in Science and Engineering 14(4), 18–25 (2012)
17. Gil, Y., Miles, S.: Prov model primer (2013), http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/
18. Green, T.J., Karvounarakis, G., Ives, Z.G., Tannen, V.: Provenance in orchestra. IEEE Data Eng. Bull. 33(3), 9–16 (2010)

19. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: PODS, pp. 31–40 (2007)
20. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers (2011)
21. Karvounarakis, G., Ives, Z.G., Tannen, V.: Querying data provenance. In: SIGMOD Conference, pp. 951–962 (2010)
22. Marinho, A., Murta, L., Werner, C., Braganholo, V., Cruz, S., Ogasawara, E., Mattoso, M.: Provmanager: a provenance management system for scientific workflows. Concurr. Comput.: Pract. Exper. 24(13), 1513–1530 (2012)
23. Moreau, L., Freire, J., Futrelle, J., McGrath, R.E., Myers, J., Paulson, P.: The open provenance model: An overview. In: Freire, J., Koop, D., Moreau, L. (eds.) IPAW 2008. LNCS, vol. 5272, pp. 323–326. Springer, Heidelberg (2008)
24. Muniswamy-Reddy, K.-K., Braun, U., David, P.M., Holland, A., Maclean, D., Margo, D., Seltzer, M., Smogor, R.: Layering in Provenance Systems. In: 2009 USENIX Annual Technical Conference, San Diego, CA (June 2009)
25. Nowakowski, P., Ciepiela, E., Harezlak, D., Kocot, J., Kasztelnik, M., Bartynski, T., Meizner, J., Dyk, G., Malawski, M.: The collage authoring environment. Procedia CS 4, 608–617 (2011)
26. Seltzer, M.: World domination through provenance (tapp 2013 keynote) (2013), `https://www.usenix.org/conference/tapp13/world-domination-through-provenance`
27. Sharman, J.L., Benson, H.E., Pawson, A.J., Lukito, V., Mpamhanga, C.P., Bombail, V., Davenport, A.P., Peters, J.A., Spedding, M., Harmar, A.J.: Nc-Iuphar. Iuphar-db: updated database content and new features. Nucleic Acids Research 41(Database-Issue), 1083–1088 (2013)
28. Wang, Y.R., Madnick, S.E.: A polygen model for heterogeneous database systems: The source tagging perspective. In: VLDB, pp. 519–538 (1990)
29. Woodruff, A., Stonebraker, M.: Supporting fine-grained data lineage in a database visualization environment. In: ICDE, pp. 91–102 (1997)
30. Zhao, J., Goble, C., Stevens, R., Turi, D.: Mining taverna's semantic web of provenance. Concurrency and Computation: Practice and Experience 20(5), 463–472 (2008)