Provenance in Databases (Tutorial Outline)

Peter Buneman
University of Edinburgh
opb@inf.ed.ac.uk

Wang-Chiew Tan[*]
UC Santa Cruz
wctan@cs.ucsc.edu

ABSTRACT

The provenance of data has recently been recognized as central to the trust one places in data. It is also important to annotation, to data integration and to probabilistic databases. Three workshops have been held on the topic, and it has been the focus of several research projects and prototype systems. This tutorial will attempt to provide an overview of research in provenance in databases with a focus on recent database research and technology in this area. This tutorial is aimed at a general database research audience and at people who work with scientific data.

Categories and Subject Descriptors

H.2 [**Database Management**]: General

General Terms

Documentation

Keywords

data provenance, lineage

1. IMPORTANCE OF DATA PROVENANCE

In computer science, provenance – also called lineage and pedigree – describes the source and derivation of data. For artifacts, the importance of provenance has long been recognized because it "can help to determine the authenticity of a work, to establish the historical importance of a work by suggesting other artists who might have seen and been influenced by it, and to determine the legitimacy of current ownership" [14]. Electronic or digital artifacts are no different; a record of provenance is essential to the trust one places in data. Many scientific data sets are the result of complex analyses or simulations. Keeping a complete record of how the computation was performed is essential: (a) to ensure repeatability, (b) to catalog the result, (c) to avoid duplication of effort, and (d) to recover the source data from the output data. In curated databases, data elements are often copied from one database to another. A knowledge of where a data element has come from is essential in assessing the quality of the database.

In addition, provenance has recently been shown to be important to understanding the transport of annotation in database views, to data integration, to view update and maintenance, and to probabilistic databases.

In this tutorial, we shall (1) provide a general overview of provenance, (2) describe some earlier research on provenance and show in detail how provenance has been used in various database applications, (3) show some emerging connections between provenance models and other areas of database research, and (4) discuss some requirements of what make good provenance recording a reality.

2. TUTORIAL OUTLINE

2.1 Part 1: Overview of Provenance

We describe two general approaches that have been used in recording provenance. The most general approach to provenance is one in which one records a complete history of the derivation of some data set. This is called *workflow* or *coarse-grain* provenance. This may involve not only tracking the interaction of programs, but also the involvement of external devices such as sensors, cameras or other data collecting equipment. It may also involve a record of human interaction with the process. A proper record of workflow provenance is essential in many scientific experiments as it enables experiments to be systematically repeated and validated by others. Some existing work in this area [18, 25, 23, 17] concentrates largely on the software architectures needed to record provenance when the components of the system are treated as black boxes. See [4, 16] for surveys that deal largely with workflow provenance.

Fine-grain provenance, the focus of this tutorial, is an account of the derivation of *part* of the resulting data set. For example, if the resulting data set is a relational database, the fine-grained provenance of a tuple in the relational database could be a tuple or a data element in the source. Why should one be concerned about fine-grained provenance when it is – presumably – derivable from the workflow? There are several reasons.

- In most cases, the entire workflow may be extremely complicated, but the derivation of the component of interest may have a simple explanation. For example, it may simply have been copied from somewhere.
- The whole workflow may not be available.
- The simplest characterization of the "workflow" may be the log or record of actions on individual components of the database. This is true of manually curated databases, which we shall discuss later.

Within fine-grain provenance, there is a further important distinction described in [8]. This is *where*- and *why*-provenance. In many cases, the effect of a query is simply to copy a data element from

[*] Supported in part by NSF CAREER Award IIS-0347065 and NSF grant IIS-0430994. Work partially done while at IBM Almaden.

SIGMOD'07, June 12–14, 2007, Beijing, China.
ACM 978-1-59593-686-8/07/0006.

some source to some target database. *Where*-provenance is simply the identification of the source elements where the data in the target is copied from. In *why*-provenance, one keeps, in addition, the justification for the element appearing in the output. To see the difference, consider the following SQL query over two relations Emp(ssn, name, deptid) and Dept(id, dname).

```
select   Emp.name, Dept.dname
from     Emp, Dept
where    Emp.deptid = Dept.id
```

Assuming that (Kim, CS) is in the result of executing the query, the where-provenance of "Kim" is the name attribute of some Emp tuple (whose value is "Kim"). The identity of which Emp tuple whose value is "Kim" is made precise by the why-provenance of (Kim, CS). The why-provenance of (Kim, CS) involves (1) the SQL query, and (2) a tuple from Emp and a tuple from Dept with the following properties: They agree on the deptid value (i.e., they satisfy the `where` clause of the query), the name attribute of the Emp tuple is "Kim" and the dname attribute of the Dept tuple is "CS".

2.2 Part 2: Applications of Provenance

We shall first review some research work on provenance in the 1990s [21, 24], as well as early 2000 [11, 8].

Wang and Madnick [21] proposed the polygen model and algebra where results of queries can carry along source attributions (i.e., provenance as a form of annotations) in each column of each tuple. The polygen algebra was not formally studied but it has inspired several subsequent research work (e.g., [9, 19, 2]). Woodruff and Stonebraker [24] first proposed the idea of building the capability of retrieving fine-grained provenance into a database management system when results of queries are not annotated with provenance. Their idea was to allow the programmer to define *weak inverses* for the functions defined in their code. Intuitively, a weak inverse, when applied to some data element in the result of a function returns some approximation to the provenance that is associated with the function. A separate verification phase is used to verify the information returned by a weak inverse. Cui et al. [11] studied the problem of computing provenance (without using weak inverses) by analyzing the operations of the relational algebra and its extensions. The work of [11] is later re-examined in [8], in the context of a tree data model.

After reviewing these research work, we examine some applications of provenance models and provenance recording (described below) that were made in subsequent years. The first two are examples of systems that simply record provenance for the reasons mentioned above – as an essential component of data quality.

Systems that record workflow provenance. Although this is not the focus of this tutorial, we briefly mention a few relevant systems that record workflow provenance. The systems we overview are from recent work on workflow provenance [4, 18, 25, 23, 17].

Curated databases. We describe how provenance has been used in manually curated databases. In particular, we review one recent work [6] that record fine-grained provenance using some simple compression techniques. They have also shown that the space overhead for doing so is acceptable.

Annotation. Annotation of data is the process of adding to or "marking up" existing data, sometimes in a *ad hoc* fashion. One would like annotations to be propagated from source to output in a systematic way and one application is to capture provenance, as in [21]. In some applications such as [3], it is also desirable to have annotations propagated from output to source. We review a few recent research that studies various theory and systems issues associated with propagating annotation of data from the source to the output, based on provenance [9, 19, 2, 12].

Probabilistic databases. Here, we review a recent application of provenance to probabilisitic databases (also known as databases with uncertainty) [1]. Provenance was used to determine whether the sources of tuples in the result of a query are independent. It was shown that the provenance of tuples can help correctly capture the set of possible instances in the result of a probabilistic query. This research was done as part of the Trio project [22], where the goal is to manage data, provenance of data and uncertainty of data as one integrated system.

Data Sharing and Data integration. Provenance has also been used in Orchestra [15, 20], a collaborative data sharing system to describe trust policies. Updates are associated with provenance information to allow the system to prioritize updates. Provenance has also been used to describe relationships between source and target data in a data exchange or integration scenario, for the purpose of understanding and debugging the specification of the integration system [10].

2.3 Part 3: Other emerging applications

In this part of our tutorial, we describe some emerging connections between provenance and other areas of database research.

Updates. The update sublanguage of SQL is normally regarded as theoretically uninteresting because it only expresses transformations that are already expressible in SQL's query language. However, when viewed as languages that also manipulate provenance, update languages are more expressive [7]. Given a table $R(A, B)$ with tuples $\{(1, 2), (8, 9)\}$, consider the following SQL expressions:

```
select *            update R        delete from R
from R              set B = 5       where A = 1;
where A <> 1        where A = 1     insert into R
union                               values (1,5)
select A, 5 as B
from R
where A = 1
```

All three expressions produce the same "result", but they all differ on the provenance that one would naturally ascribe to the tables, tuples and data values in the source. Can we use provenance in the analysis of update languages? Also, suppose we use one query to define a transformation of the database and another, independent, query to describe the provenance associated with this transformation. That is, the provenance is explictly defined by a query. Under what circumstances is explicit provenance captured by the implicit provenance semantics of an update language?

Trust in data sharing and view maintenance. Provenance is central to trust in data sharing and integration, but how can trust be quantified or formally described? Recent work on provenance semirings [13] models provenance as semirings of polynomials. It is a promising approach for describing trust and may also be useful in reasoning about recursive view maintenance.

Workflow provenance. Returning to workflow provenance, some interesting recent work [5] breaks the "black-box" assumption in workflow provenance on streaming data, giving a form of fine-grain provenance in a temporal dimension. For example if we know that a sliding window computation is being performed, we know that a value in the output stream depends only on a segment of the input stream. Conversely, it should be possible to extend accounts of where-provenance in query languages to deal with aggregating functions. For example, a data value produced by a sum in a SQL group-by depends on a limited number of values in a column on the source language. If a common framework for stream languages and query languages can be found, it may be possible to unite the two notions of provenance.

3. PROVENANCE IN CONTEXT

Time permitting, we shall also discuss what is needed to make good provenance recording a reality.

- **Data models.** How do we prepare our databases so that provenance data is intrinsic to the schema rather than an external annotation?
- **Archiving.** Databases change. If we are recording the provenance of a data element that was derived from some database, then it is important to keep the state of the database at the time the derivation took place.
- **Capturing user behavior.** In curated databases, data elements are often entered by copying from some web page and pasting it into some other web or forms interface to a database. This action takes place through the user interface, and it is typically where provenance information is lost. How do we improve these interfaces to be "provenance-aware"?
- **Extending query languages.** As we have seen, queries that are equivalent in that they produce the same output may not be equivalent in the provenance they convey. Do we need more sophisticated query languages, or should we be content with annotating existing query languages?
- **Aggregate and non-monotonic queries.** Most research work on provenance have focussed on queries that do not involve negation and arbitrary functions. What provenance information should be captured for such queries?

About the presenters

Peter Buneman is Professor of Database Systems at the University of Edinburgh. He has worked in several areas of databases including query languages, semistructured data and, recently, a number of issues in scientific data.

Wang-Chiew Tan is an Assistant Professor at the Computer Science Department at University of California, Santa Cruz since September 2002. Her current research interests include data provenance, annotations, archiving, and information integration.

4. REFERENCES

[1] O. Benjelloun, A. D. Sarma, A. Y. Halevy, and J. Widom. ULDBs: Databases with Uncertainty and Lineage. In *Very Large Data Bases (VLDB)*, pages 953–964, 2006.

[2] D. Bhagwat, L. Chiticariu, W.-C. Tan, and G. Vijayvargiya. An Annotation Management System for Relational Databases. *Very Large Data Bases (VLDB) Journal*, 14(4):373–396, 2005.

[3] biodas.org. http://biodas.org.

[4] R. Bose and J. Frew. Lineage Retrieval for Scientific Data Processing: A Survey. *ACM Computing Survey*, 37(1):1–28, 2005.

[5] S. Bowers, T. McPhillips, B. Ludäscher, S. Cohen, and S. B. Davidson. A Model for User-Oriented Data Provenance in Pipelined Scientific Workflow. In *International Provenance and Annotation Workshop (IPAW'06)*, Chicago, Illinois, 2006.

[6] P. Buneman, A. Chapman, and J. Cheney. Provenance Management in Curated Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 539–550, 2006.

[7] P. Buneman, J. Cheney, and S. VanSummeren. On the Expressiveness of Implicit Provenance in Query and Update Languages. In *International Conference on Database Theory (ICDT)*, pages 209–223, 2007.

[8] P. Buneman, S. Khanna, and W.-C. Tan. Why and Where: A Characterization of Data Provenance. In *International Conference on Database Theory (ICDT)*, pages 316–330, 2001.

[9] P. Buneman, S. Khanna, and W.-C. Tan. On Propagation of Deletions and Annotations Through Views. In *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PODS)*, pages 150–158, 2002.

[10] L. Chiticariu and W.-C. Tan. Debugging Schema Mappings with Routes. In *Very Large Data Bases (VLDB)*, pages 79–90, 2006.

[11] Y. Cui, J. Widom, and J. L. Wiener. Tracing the Lineage of View Data in a Warehousing Environment. *ACM Transactions on Database Systems*, 25(2):179–227, 2000.

[12] F. Geerts, A. Kementsietsidis, and D. Milano. MONDRIAN: Annotating and Querying Databases through Colors and Blocks. In *International Conference on Data Engineering (ICDE)*, page 82, 2006.

[13] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance Semirings. In *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PODS)* (To appear), 2007.

[14] Harvard University Art Museums, Provenance Research. http://www.artmuseums.harvard.edu/provenance/, cited on 14 November 2006.

[15] Z. Ives, N. Khandelwal, A. Kapur, and M. Cakir. Orchestra: Rapid, Collaborative Sharing of Dynamic Data. In *Conference on Innovative Database Systems Research (CIDR)*, 2005.

[16] Y. Simmhan, B. Plale, and D. Gannon. A Survey of Data Provenance in E-Science. *SIGMOD Record*, 34:31–36, 2005.

[17] Y. L. Simmhan, B. Plale, and D. Gannon. A Framework for Collecting Provenance in Data-Centric Scientific Workflows. In *International Conference on Web Service (ICWS)*, 2006.

[18] M. Szomszor and L. Moreau. Recording and Reasoning over Data Provenance in Web and Grid Services. In *International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE)*, pages 603–620, 2003.

[19] W.-C. Tan. Containment of Relational Queries with Annotation Propagation. In *Database Programming Languages (DBPL)*, pages 37–53, 2003.

[20] N. E. Taylor and Z. Ives. Reconciling while Tolerating Disagreement in Collaborative Data Sharing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 13–24, 2006.

[21] Y. R. Wang and S. E. Madnick. A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective. In *Very Large Data Bases (VLDB)*, pages 519–538, 1990.

[22] J. Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. In *Conference on Innovative Database Systems Research (CIDR)*, pages 262–276, 2005.

[23] S. C. Wong, S. Miles, W. Fang, P. Groth, and L. Moreau. Provenance-based Validation of E-Science Experiments. In *Proceedings of Internation Semantic Web Conference (ISWC)*, pages 801–815, 2005.

[24] A. Woodruff and M. Stonebraker. Supporting Fine-grained Data Lineage in a Database Visualization Environment. In *International Conference on Data Engineering (ICDE)*, pages 91–102, 1997.

[25] J. Zhao, C. Wroe, C. Goble, R. Stevens, D. Quan, and M. Greenwood. Using Semantic Web Technologies for Representing e-Science Provenance. In *International Semantic Web Conference (ISWC)*, pages 92–106, 2004.