2nd USENIX Workshop on the Theory and Practice of Provenance (TaPP '10)

San Jose, CA
February 22, 2010

INVITED TALK

- ***Naming, Identity, and Provenance***
  Jim Waldo, Distinguished Engineer, Sun Microsystems Laboratories

  Summarized by Aditya Parameswaran (adityagp@cs.stanford.edu)

Jim Waldo's talk centered on identity as a philosophical notion and how it relates to how we think about provenance and data.

Waldo discussed what "identity" means. Identity is hard to describe (how do we say that a is the "same" as b?), understand (if a is the same as b, what does that mean?), and teach. In addition, identity may be discovered through references. For example, two items that were called by different names might be discovered to be the "same."

Our notion of what is the "same" is also confusing. To illustrate this, Waldo described the paradox of the Ship of Theseus: There is an original Ship of Theseus, whose parts are stripped off and used to build a new identical ship. At the same time, the old ship is embellished with new parts to replace the old parts. At this point, it is hard to say which of these two ships is the "same" as the original Ship of Theseus.

Even in computer science, there is ambiguity in "identity." For instance, there are two notions of identity in programming languages: referential identity (== in Java) and structural identity (.equals() in Java). The name of a variable is equivalent to the reference, while structural identity actually compares the content.

Waldo spoke of the connections between the problems of reasoning about identities and modal logic. He suggested that reasoning about various "versions" of an object can be done by means of possible worlds, with the notion of a "Designator," i.e., a canonical version of an object, whose properties may change over various possible worlds.

In science, there is a need for reproducibility in experiments, in order to get the "same" result. However, reproduction is hard, since in the worst case, one might need a snapshot of the entire universe to reproduce the same environment for the experiment. Thus the challenge is to maintain the "right stuff" in order to be able to reproduce experiments to a rough approximation. Traditionally, we do this in computer science by maintaining data via version control. However, it is still unclear what we need to save (as provenance) to ensure reproducibility.

Waldo then cautioned us that identity is a deep unsolved philosophical problem which has been around for centuries, and thus it is likely that it will not be solved in the near future. However, he suggested that for special cases, understanding and solving the problem of identity should be possible.

SECURITY AND EXPERIENCE

- ***Trusted Computing and Provenance: Better Together (long paper)***
  John Lyle and Andrew Martin, Oxford University Computing Laboratory
- ***Towards a Secure and Efficient System for End-to-End Provenance (short paper)***
  Patrick McDaniel, Kevin Butler, and Stephen McLaughlin, Pennsylvania State University; Radu Sion and Erez Zadok, Stony Brook University; Marianne Winslett, University of Illinois at Urbana-Champaign
- ***Towards Query Interoperability: PASSing PLUS (long paper)***
  Uri J. Braun and Margo I. Seltzer, Harvard School of Engineering and Applied Sciences; Adriane Chapman, Barbara Blaustein, M. David Allen, and Len Seligman, The MITRE Corporation

- ***Provenance Artifact Identification in the Atmospheric Composition Processing System (ACPS) (short paper)***
  Curt Tilmes, NASA Goddard Space Flight Center and University of Maryland, Baltimore County; Yelena Yesha and Milton Halem, University of Maryland, Baltimore County

No reports are available for this session.

INVITED TALK

- ***Provenance for the Nationwide Health Information Network***
  Latanya Sweeney, Distinguished Career Professor of Computer Science, Technology, and Policy, CMU, Director of the CMU Privacy Laboratory, and Visiting Scholar at the Harvard Center for Research on Computation and Society

Summarized by Robin Smogor (pyrodon@gmail.com)

Recently, national funding is pushing the creation of a nationwide health information network. Currently hospitals and care facilities are not sharing information even locally, due to privacy concerns, except for billing or claims which are forwarded to national databases in the various insurance companies. Building a system to provide the attributes desired by policymakers and health care providers involves tracking many different kinds of provenance, and solutions that naively use one kind will often cause issues in other kinds. The provenance community needs to rise to the design challenge soon in order for a solid, good network to become adopted as hospitals move towards sharing information nationwide.

Someone pointed out that when digging into complex provenance a little, we sometimes want a cumulative answer and also proof but are not allowed to share proof. For example, we want to count the number of newly diagnosed cases of HIV in an area, but providers can't share identifiable characteristics that are needed for deduplication. Sweeney agreed and said that this is one of the big problems we need to solve. Even negative information can reveal information (no new cases in a hospital gives information about the site). Sweeney also mentioned that part of the 2009 stimulus bill in the US called for nationwide electronic medical records by January 2011. That deadline will likely be postponed but will eventually happen, and if we don't propose a better solution to the committee distributing the money and the various projects working towards nationwide EMR, they will default to using social security numbers as the unique patient identifiers, which would not be good. Someone else commented that we want national sharing of case details in some form, especially for rare diseases. You really want clinicians to have access to good information so they can treat things they have never seen.

Someone else asked about an architecture that focused on the patient, having each control their own flash drive. Sweeney answered that providers consider it their information, not the patient's. Providers don't want to give you full access to your own medical records because it might "confuse you." There is also the liability aspect—they don't trust the patient to protect their own data. Another person pointed out that all labs are not created equal. The local lab may use less accurate machines or methods than the regional lab. Sweeney responded that the value is in the report, not the processes or materials. That is, a PCP can't read an x-ray any better than we can. The value is in the radiologist's report and trust is in the radiologist, not in the lab tech who took it.

Someone else asked about policy, pointing out that medical records are in C42 format, which does not include provenance. Since provenance is not covered by the standard, how can we get it included? Sweeney answered that clinical information is mostly in plain text, not a database format, for anthropological reasons.

Finally, an attendee wondered about modeling the health network on credit reporting, with three approved competing businesses. Sweeney said she liked the idea of health reporting agencies, but it's not getting support right now, even though there's a nice proof of concept. The government is paying from the bottom up, and medical software manufacturers are doing a big turf sweep, tying up California.

SYSTEMS AND USES OF PROVENANCE

Summarized by Peter Macko (pmacko@eecs.harvard.edu)

- ***Panda: A System for Provenance and Data (short paper)***
  Robert Ikeda and Jennifer Widom, Stanford University

Panda is a work-in-progress project developing a complete, general-purpose solution for capturing, storing, and querying provenance. The project focuses on workflow-based systems and captures both provenance and data, which enables it to support a rich set of features. For example, a user would be able to pick one of the inputs and trace it through the computation, or select a piece of the output and trace it backwards. The system would also be able to propagate a change in the input by recomputing only the parts of the workflow affected by the change. Similarly, the system would be able to check whether a given result is still valid after correcting an input and then use this forward propagation method to refresh its value.

One of the goals of Panda is to seamlessly support relational operators with known, well-defined semantics as well as fully opaque operators. The system would further support query-driven provenance collection (record only the provenance that you need to answer pre-specified queries), lazy provenance computation and storage (compute provenance of selected parts of a workflow only when needed), multiple granularities of provenance, and approximate provenance (allow the system to record provenance imprecisely in order to save space).

A member of the audience asked the speaker how the provenance is captured—whether the system places wrappers around the workflow operators or executes them inside

a provenance-aware interpreter. The speaker explained that they do not use a specialized Python interpreter; the individual workflow operators export their own provenance back to the system.

- ***Towards Practical Incremental Recomputation for Scientists: An Implementation for the Python Language (long paper)***
  Philip J. Guo and Dawson Engler, Stanford University

Scientific computations run typically on the order of minutes to hours. This makes the development cycle unacceptably long: after a developer corrects a few lines of code at the end of the program, he or she has to rerun the entire computation. Most developers thus break their programs into small pieces, which read and write intermediate results. While this reduces the development cycle, it greatly increases the complexity of the code, is time-consuming, and introduces many new bugs.

The paper describes a system that addresses this problem by providing a modified Python interpreter that automatically memorizes saves) results of functions. The paper focuses specifically on programs written in Python, but its approach generalizes to any interpreted general-purpose imperative language. The system detects code changes both in the actual memorized function and in all functions it calls. The interpreter also keeps track of which functions read which files and on the state of the global variables that the function reads for any given memorized result. Furthermore, the system is careful not to memorize the results of impure functions, which mutate non-local values, write to files, or call non-deterministic functions.

The described approach uses only dynamic analysis, so some members of the audience were wondering about the possibility of using static analysis. The speaker explained that using dynamic analysis is conceptually more straightforward, but it is possible to use static analysis as an optimization. Furthermore, static analysis is difficult in interpreted languages with no explicit types, such as Python. Another member of the audience asked whether the system influenced the way its users develop their programs. The authors did not come far enough to provide their system to its intended real users, but ideally, the users would structure their programs using more self-contained functions.

Why did they choose to use Python for their work? One of the main reasons was the authors' personal familiarity with this language, but this technique should work with any other high-level dynamic language, such as Matlab. When does the system purge its memorization cache? They remove entries from the memorization table whenever the system detects a new version of the code. What about the space overhead and about dealing with changes in libraries? There is anecdotal evidence that the space overhead depends on the size of the intermediate data and that the changes in libraries would be detected by the interpreter's code change-detection mechanism. Finally, in the response to a related question, the speaker explained that the system does not yet handle function calls outside Python.

- ***Using Provenance to Extract Semantic File Attributes (short paper)***
  Daniel Margo and Robin Smogor, Harvard University

The authors present a method for automatically extracting meaningful semantic attributes of files from their provenance, or more precisely, the context in which they are used. For example, if an application always reads a file in its directory, it is most likely its component, but if the application sometimes writes a file outside its directory, it is probably a document.

The described tool captures provenance from PASS, collapses versions of the same objects into single nodes, and then proceeds with feature extraction. The program produces multiple ancestor and descendant graphs for each file with different features, such as with collapsed nodes with the same name or path, or with just file or process objects. The program then extracts simple per-file statistics, such as node and edge counts in the neighborhood of each file. The authors also experimented with graph clustering and other sophisticated methods of feature extraction, but they did not produce good results.

The authors next combined the extracted features with relevant meta-data of existing files collected using the stat command, and then constructed a decision tree. They evaluated their approach by predicting file extensions (because that makes it easy to establish ground truth) and achieved 86% accuracy.

Which features did the decision tree split on? It typically split on the depth of the provenance graph, because this is an indication of how often the file is accessed. A good research direction is to consider the shape of the graph. Do semantic attributes reflect what the document contains? This is still a research question; another question is whether the usage of the file reflects what the user thinks about the file. So far, the project has shown that the way a file is used predicts its type, and it is an open question how far it is possible to push this.

Is their method an alternative for content-based extractors? It is beneficial to extract as many rich semantic attributes as possible, so this method should be used in conjunction with traditional content-based extractors. Furthermore, this method still allows you to extract attributes from files that were already deleted. Finally, someone suggested that the authors should consider expanding their work to include feature extraction methods from graph indexing and querying literature.

MODELS: NEW AND DIFFERENT WAYS OF THINKING ABOUT AND REASONING ABOUT PROVENANCE

Summarized by Abhijeet Mohapatra (abhijeet@stanford.edu)

- ***A Graph Model of Data and Workflow Provenance (long paper)***
  Umut Acar, Max-Planck Institute for Software Systems; Peter Buneman and James Cheney, University of Edinburgh; Jan Van den Bussche and Natalia Kwasnikowska, Hasselt University; Stijn Vansummeren, Université Libre de Bruxelles

James Cheney presented a graphical model that captures the common formalism for workflow and database provenance. Cheney's work addresses the fact that workflow systems are seldom accompanied by formal specifications of the desired provenance semantics. Hence, it is difficult to integrate database and workflow provenance or compare provenance generated by different systems.

Cheney proposed a model based on provenance graphs that document the evaluation of a DFL program. Such graphs contain values as well as evaluations. Ignoring the value structure in the provenance graph would produce the order of evaluation of processing nodes.

Cheney described their implementation of the proposed graphical model in Haskell. He then discussed how different provenance queries could be expressed over provenance graphs. These queries were a mixture of Datalog and annotation propagation queries. Most of the queries related to where and why provenance in databases. Finally, he outlined some unsolved problems that relate to modeling updates to provenance graphs and identifying classes of provenance queries that exhibit symmetry in querying the provenance graph "forward" vs. "backward."

- ***A Conceptual Model and Predicate Language for Data Selection and Projection Based on Provenance (long paper)***
  David W. Archer and Lois M.L. Delcambre, Portland State University

David Archer presented a predicate language that supports a broad class of provenance queries having applications in data curation. Current provenance models have two major shortcomings. First, they are either fine-grained or coarse-grained. Second, annotation management in such systems is messy. Thus, there is a need to develop a language that helps end users pose queries that select data by its provenance information.

Archer described a conceptual model for capturing provenance that separated provenance tracking and its manipulation by end-users. He then proposed a predicate language to record provenance for SELECT and PROJECT operators using "path qualifiers."

Archer later evaluated the proposed model against Trio and PASS's provenance models comparing the expressivity of provenance queries.

- ***On the Use of Abstract Workflows to Capture Scientific Process Provenance (long paper)***
  Paulo Pinheiro da Silva, Leonardo Salayandia, Nicholas Del Rio, and Ann Q. Gates, University of Texas at El Paso

Paulo Pinheiro da Silva presented a model to capture and reuse how provenance in scientific processes. He was prompted by the fact that scientists often track provenance without using methods specifically designed to record provenance, and this makes it hard to reuse the recorded provenance. In his proposed model he used Process Markup Language (PML) to encode distributed provenance.

Da Silva began his talk by describing the languages and tools commonly used to capture provenance of scientific processes. He later described how provenance could be captured for automated as well as manual processes. He also outlined a data annotation scheme to support provenance queries.

At the end of the talk, da Silva noted that the proposed approach to capture provenance might not be scalable.

- ***Provenance-based Belief (short paper)***
  Adriane Chapman, Barbara Blaustein, and Chris Elsaesser, The MITRE Corporation

Adriane Chapman presented a mechanism to express trust in data sources without actually accessing them. This is intended to help answer provenance queries based on a certain level of trust. She commented that provenance graphs can be viewed as a causal structure which can be used to compute belief of an output from assessments of input data and derivations.

Chapman talked of integrating Bayesian causal reasoning and provenance. She described how belief of outputs could be computed by generating conditional probability tables for the output tuple's intermediate derivations. She commented that the provenance store could be used to identify sharing between sources. Modeling provenance with a causal model would enable propagation of beliefs based on shared and independent sources.

Chapman ended the talk by saying that her group is currently implementing the causal model to capture provenance into a real system for evaluation purposes.