Application of Provenance for Automated and Research Driven Workflows

Tara Gibson, Karen Schuchardt, and Eric Stephan

Pacific Northwest National Laboratory,
P.O. Box 999, Richland, WA, USA
{Tara.Gibson,Karen.Schuchardt,Eric.Stephan}@pnl.gov

Abstract. Provenance has recently become a popular topic for workflow execution environments but it is also relevant to other applications, such as long-running, user-driven "research workflows", problem solving environments, and data streaming (data analysis) environments. This paper presents a number of use cases where provenance can play an important role in understanding how data was derived, how decisions were made, and enable sharing of data from a variety of sources. We break down the requirements elucidated by our use cases and discuss our experiences in applying an existing provenance system to these use cases.

Keywords: provenance, metadata, use cases, workflow.

1 Introduction

Science relies on intermediate results to guide future research; every result needs to be supported with detailed records of its derivation (provenance). At the same time, science is becoming more complex, involving both manual and automated pipeline processing steps, and consequently records are more difficult to maintain. Adding to this complexity, simulations and experiments often need to be analyzed together. This creates a challenge for scientists who need integrated views of the different types of provenance records associated with computation, experimentation, and analysis. The existing method for capturing data derivation, traditional laboratory notebooks, suffers from the mismatch between media (paper versus electronic). Electronic laboratory notebooks fill an important niche but are often not used and usually capture only a manual record of research within a desktop publishing type environment. Our vision is to go one step further, automatically providing a very detailed recording of any complex scientific processes.

Documenting the scientific process requires an architecture designed to record, store, manage, and access the provenance describing the scientific process [1]. While the topic of provenance has gained interest in the computer science community it is at a relatively early stage of investigation and application. For example, Simmhan, et. al [2], recently surveyed the use of provenance in e-science applications. Based on the projects surveyed, provenance is typically implemented as an application extension,

J. Freire, D. Koop, and L. Moreau (Eds.): IPAW 2008, LNCS 5272, pp. 128–135, 2008.

rather than a component that can be applied to many systems. Chapman et al. [3] provided a useful set of general requirements based on tracking provenance while combining and manipulating protein interaction data. Based on these requirements as well as those described by Miles et al. [1] we define a set of requirements which apply to a provenance architecture. Many of these requirements were drawn from automated workflow implementations with an emphasis on a Service Oriented Architecture (SOA) architecture. To complement this research, we explore the requirements gathered from use cases related to both automated and user-driven workflows and data systems. From these motivating use cases, we introduce new requirements for consideration while confirming many existing requirements. In the remainder of this paper, we present our use cases (section 2), describe the requirements resulting from our analysis (section 3), and we compare our findings to other state-of-the-art provenance architecture requirements.

2 Use Cases

The following use cases are derived both from direct experiences on projects that have provenance needs and from conducting interviews with scientists across diverse domains. We present the use cases in two categories: automated workflow, and user-driven research workflow. Each use case is described in a moderate amount of detail and is accompanied by a summary of the applicability of provenance for that context. Requirements are presented as a synthesized list in the next section.

2.1 Automated Workflow

U1. Sensor Analysis Pipelines. Intrusion detection systems currently have production network sensors deployed at 30 sites around the country. These sensors monitor a combined network data volume of approximately 30 TB per day for malicious activity. Attacks must be detected as they occur in order to stop intruders from gaining entry and damaging sensitive systems. To enable this detection, aggregation and summarization techniques are applied to compress the data streams into manageable volumes. A component-based (SOA-based) messaging and integration architecture for creating analytical pipelines is used to provide anomaly detection and analysis by following a general pattern of processing stages; ingest, aggregation, signature generation, anomaly detection, context analysis, and data visualization.

The anomaly detection stage is responsible for using statistical and heuristic methods for automatically determining whether a given signature should be considered a significant event. As algorithms are improved, the pipeline may be rerun on the same data to compare the affects of the changes on the identification of significant events. In this context, provenance can be used to review incidents, understand why they were marked as significant, and analyze the impact of changes to the algorithms. This is a different model of provenance capture for workflow than others we are familiar with in that it becomes important to save provenance only for notable events. Additionally, due to the volume of data processed, the provenance capture mechanism must not incur significant delay in the pipeline performance.

U2. Predictive Biology. System's Biology research relies on the collection and analysis of massive amounts of complex biological response and genetic data with the goal of identifying signatures that define, or are predictive of, biological systems and their response to perturbation. Automated workflow is used for gene set enrichment calculations using KEGG pathways and Gene Ontology (GO) terms. A second workflow is being developed to automate the quality control and normalization of Microarray data. This process retrieves data from an external source, and sends it to a machine capable of performing the statistical calculations using R scripts.

In these contexts, provenance allows users to validate the results of past workflows and examine the full derivation of a result. Additionally, since public data sources such as KEGG and GO are constantly changing as new research is added and curated by the community, provenance can track the information about the data source (such as version) to understand differences in results that occur over time. Provenance can also be used to determine which analysis have been run and on which data sources/versions.

U3. Protein Interaction Discovery. Proteins have largely been studied as independent units of function. However, most proteins cooperate with one another in the form of higher level functions, referred to as protein complexes or pathways. A deeper understanding of protein interactions help scientists understand how proteins work together. Protein interaction databases play a vital role in helping identify protein complexes that may ultimately share the same characteristics in two different organisms. The hope is that by discovering characteristics within one organism, searching protein interaction in another organism may lead to similar types of behaviors.

Biologists continually go through a painstaking and time consuming process to manually access distributed data sources, merge the data sets in some way, and perform analyses on the data. For example, the question "*What proteins correspond to genes that are up-regulated at 3hr and 4hr in my microarray data, and which proteins are they known to interact with?"* illustrates the need to correlate experimental results derived from a microarray experiment and compare them to various public protein interaction databases. Answering this question requires several major steps suitable to workflow automation techniques. In this case, provenance can again be used to compare workflow executions to understand the impact of data or algorithm changes on the final result. To do so, it is useful to capture intermediate query results that can vary based on database version, and have an effect on later steps in the workflow.

2.2 User-Driven Research Workflow

We apply the term research workflow to a group of use cases where the goal is to document data derivation and provenance of long-running and at times ad-hoc user-driven research activities.

U4. Subsurface Modeling. Subsurface modeling employs computerized mathematical models to explore complex physical systems that cannot be easily or cost effectively investigated through experiment. When applied to environmental remediation, the goal might be to model processes to understand how contaminants react and move through the environments. Developing this understanding necessarily involves running numerous (tens or hundreds of) related simulations. The research process typically involves

running a small number (often one) of simulations, analyzing results and deciding what to do next. There is usually a derivation relationship among the simulations; that is, a researcher will explore along one line of investigation, then go back and explore along another line perhaps branching multiple times within a given line of investigation. A key aspect of this process is that there are many branches of investigation with complex relationships between simulations and across branches.

In current practice, this type of study is maintained in a directory structure with simple metadata naming conventions on directories. However the relationships are not tracked. Weeks later, the detail of the relationships between simulations becomes difficult to recall, even by the researcher performing the simulations. To collaborators, it is undecipherable. In this context, provenance can be used to record the complex relationships between simulations, document branches of investigation, and enable a user to organize and understand their overall process. It can also be used to decide on next steps, present customs views that reduce the complexity, repeat a sequence of steps with different initial conditions, and search for simulations based on detailed context information typically found only in the data files.

U5. Comparative Analysis. The study of complex computational biology and computational chemistry simulations is pursued with the goal of improving the understanding of complex protein interactions. Scientists make extensive use of several high-performance computational tools to produce and analyze data and ultimately to design protein-based scaffolds for environmental cleanup. This can, in principle, be achieved by performing a wide array of simulations of several protein variants under a variety of physico-chemical conditions (pH, temperature, ionic strength). Data for published work must typically be retained for five years.

The research workflow for this problem contains few steps, but numerous simulations and iterations. Once a collection of simulations is complete, visualization tools are used to analyze candidate simulation trajectories exhibiting particular behavior under a variety of conditions. Based on the chosen candidate trajectories, provenance can be used to answer important questions such as: what simulations were used in my comparative analysis, and what simulations and analysis were used in my cited research. Additionally, a researcher will want to quickly access important summary information about simulations (who ran them, under what conditions) and analyses (why was a particular line of investigation followed), and gain direct access to the data files.

U6. Archive Data Mining and Sharing. The Environmental and Molecular Sciences (EMSL) facility houses a variety of high performance experimental and computational resources dedicated to environmental molecular sciences research. The facility has an archive with hundreds of terabytes of data essentially treated as a large file system. There is growing recognition of the value of documenting the data to improve overall effectiveness of the facility. Early versions of the archive required and enforced the use of metadata. However, this requirement was too onerous and effectively discouraged use of the archive. Collection of metadata must therefore be lightweight, customizable, and optional. Further, it is often important to track the relationships between experiments or between experiments and computations. For example, Nuclear Magnetic Resonance (NMR) experiments may be run on samples and computer models used to

determine the 3D structure. It is desirable to capture those relationships in searchable, browse-able, notebook type form.

In this context, provenance and metadata can be used to answer important questions such as: what experiments have already been run and under what conditions, what research has been conducted on particular organisms, by whom, with what equipment, what data is available, and how is it related to other experiments?

Harvesting technology is ideal in such an environment, particularly if it can be readily customized and if mechanisms for describing relationships are provided.

3 Use Case Findings

From these use cases, we derived a set of requirements, both functional and non-functional. The full derivation of these requirements, which is beyond the scope and available space for this paper, is based on our detailed discussions and experiences. However, we cross reference the connections between them in the list below and highlight what we view as new requirements and other unique aspects of our findings.

R1. *Record (and query) arbitrary information about individual processes, data that moves between processes, and the relationships between them (U1-U6)*

R2. *Record enough information to enable references of data regardless of size or location (U2-U6)*

R3. *Extract and record customized file metadata for context searching (U4, U6)*

R4. *Record only the provenance from significant events and the processes and data that led to the identification of the event (U1)*

R5. *Identify processes, experiments, or data as a collection of related work and allow users to record arbitrary annotations and define new relationships. (U1-U6)*

R6. *Record provenance of high throughput pipelines with minimal impact on performance (U1)*

R7. *Determine who ran a particular process, under what conditions, and which settings were used (U1-U6)*

R8. *Determine if an analysis or experiment has previously been run (U2, U6)*

R9. *Identify data generated from a particular process (U1-U6)*

R10. *Retrieve information to be presented for application specific views (U2, U3)*

R11. *Identify contextual information and results from access to dynamically changing data sources and versions used in an analysis (U2, U3)*

R12. *Examine full derivation of the result or significant event (U1-U3)*

R13. *Determine where a process/data was used for data that should be regenerated due to an algorithm or data source change (U2)*

R14. *Query for derivation graph, filtering on level of detail (U1-U6)*

R15. *Compare multiple runs of the same workflow execution (differentiated by data source or software module versions) to analyze the effects of the changes(U1-U3)*

R16. *Retrieve process documentation to re enact an experiment or workflow using new inputs or parameters (U3-U5)*

From the list of requirements we identify the following highly abstracted core capabilities:

- Record data about process, data, relationships
- Group items together for comparison
- Record arbitrary metadata
- Standards-based search capability
- Examine process and data that led to result
- Identify the overall impact on a workflow due to changes in process/data.

Our goal in examining such diverse use cases is to verify the core capabilities applicable to virtually all use cases and to understand the extension points and design constraints necessary to support important, but non-universal capabilities. These core capabilities show great overlap with those the requirements described in Miles et. all. [1]. We also identified several requirements which we view as new additions to published capabilities. While these additions do not affect the core capabilities of a provenance architecture, they represent design considerations that impact may APIs or system design. R3 is one such case. Provenance and metadata stored within existing files (which may include who, what, when information, references to source files, and application specific contextual information), must be extracted and made accessible to satisfy query requirements. This is particularly important when provenance is applied to ad hoc research workflows where the ability to capture process information is more constrained.

Another new requirement, R4, involves storing only the significant provenance in a workflow, this applies particularly to data streaming environments. To capture all data in such an environment would essentially duplicate the original data stream and overwhelm the system both in scalable storage and query interpretation. These requirements suggest a transaction-oriented capability where a provenance record can be constructed during execution and committed only when a positive identification of significant event is made. R6 introduces the non-functional requirement for minimal overhead associated with a provenance capture mechanism in an automation environment. Critical systems processes need to proceed as efficiently as possible and provenance should not interfere with this in any way. This suggests the need for an asynchronous recording mechanism. Finally, in R14, we identify the need to reduce the data derivation graphs (at query time) to the level of detail necessary to its end purpose. A powerful view filter would support filtering based on arbitrary metadata about either process or data. The Open Provenance Model (OPM) has also made a point of providing for multiple graph descriptions or 'accounts' [4]. This is described as offering different levels of explanation for such execution, such sub graphs are also known as alternate accounts.

4 Experiences

The exploration of the described use cases and requirements presented us with challenges which we needed to adapt to a generic architecture. To meet these challenges we altered our previous architecture as described in Schuchardt, et. all [5] with several modifications, We adopted the use of RDF for its support of graph queries, arbitrary

relationships and metadata, and standards. We also developed a transaction oriented API, as needed by R4, and incorporated several of the key ideas of the OPM into our existing model. Below are a number of other challenges that we encountered while studying the use cases.

Data Overload: Even with desired view capabilities, in automated systems it is still possible to capture too much meaningless provenance, overloading the database technology. An example of this is the protein interaction discovery workflow, which was implemented on an automated workflow platform with a plug-in mechanism designed to capture provenance when any minor event occurs. We found that by ingesting every minor call within the workflow engine, that we were quickly flooded with too much detail. For this reason, workflow systems need design time control to manage the level of detail captured.

Client Side Filtering: No capabilities for server side view filtering exist at this time. For efficiency reasons, some applications thus developed client-side graph filtering capabilities to fulfill the requirement.

User Views: General tools/browsers are useful for capabilities such as simple browsing, but for most use cases, specific interfaces are required. The view desired by each application can vary greatly in level of detail or interpretation. For use cases such as subsurface modeling, a more extensive classification may be desired to represent the nuances between various actions. To resolve this, the provenance model could be extended to describe various types of actions, such as a compute job versus a data transfer. Among other things, this classification can be used to filter views where certain types of actions are more interesting to the end user and to group related research.

Language Bindings: Our initial implementation of a provenance API was in java but several of our use cases required other languages (python, C++). A general provenance system should support multiple language bindings or a standard protocol (e.g. REST) that can easily be accessed in any language. In the latter case, a language wrapper is still necessary to reduce the burden of adding provenance to a system.

Scalability: Server scalability quickly became an issue for both Sensor Analysis (U1) and Archive Data Mining (U6). To support the full number of datasets that can be encountered in either use case, we must have a storage solution that can scale to billions of triples or support queries and relationships across multiple (federated) stores.

Augmentation: We encountered numerous examples where users would like to go back and augment the provenance record. For example, when a user publishes a paper using results generated by a workflow, they will want to later go back and associate the paper with the provenance record. In simulation environments, some users want to identify and annotate processes, data or entire sub graphs with notes and analyses or manually make associations within or between different sub graphs.

5 Conclusions

Our use case studies have documented compelling examples of the benefits that provenance can provide for a diverse set of domains. Reviewing these use cases has given us a sense of similarity between nearly all workflows, and allowed us to validate a number of existing requirements of a provenance system, as well as present several new ones. To support several of these new requirements, we envision revisiting our provenance model as well as ensuring better interoperability with the OPM, we also plan to improve the API, using standardized recording protocols and adding multiple language bindings. By applying these requirements to our architecture we expect to produce an adaptable, effective system for the support of various domains.

Acknowledgement

The research described in this paper was conducted under the Laboratory Directed Research and Development Program at the Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL0 1830. The research in subsurface workflows is supported by the U.S. Department of Energy's Office of Science under the Scientific Discovery through Advanced Computing (SciDAC) program.

References

1. Miles, S., Groth, P., Branco, M., Moreau, L.: The Requirements of Using Provenance in e-Science Experiments. J. Grid Comput. 5(1), 1–25 (2007)
2. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. 34(3), 31–36 (2005)
3. Chapman, A., Jagadish, H.V.: Issues in Building Practical Provenance Systems. IEEE Data Eng. Bull. 30(4), 38–43 (2007)
4. Moreau, L., Freire, J., Myers, J., Futrelle, J., Paulson, P.R.: The Open Provenance Model. In: Luc Moreau at Workshop on Principles of Provenance, Edinburgh, Scotland, November 20 (2007)
5. Schuchardt, K.L., Gibson, T.D., Stephan, E.G., Chin, G.: Applying Content Management to Automated Provenance Capture Concurrency and Computation. Practice & Experience 20(5), 541–554 (2007)