ABSTRACT
As the volume of scientific data increases, the need for automated data provenance has expanded. Currently, several provenance systems exist to aid users in recording and querying provenance data. They range from very specific programs designed to accomplish a small number of clearly defined tasks to broad-range applications intended to appeal to a wider audience. This paper discusses several provenance systems and describes some areas for future study.

1. INTRODUCTION
Provenance means “lineage.” In antiques, provenance deals with how a piece came to be in its current condition. Data provenance is the understanding of the origins of data, as well as the transformations that the data has undergone in order to arrive at its current state [9].

A workflow is an example of what has been termed prospective provenance. Prospective provenance is the “recipe” for a procedure. It defines the steps that should be taken in order to accomplish some goal [7]. This contrasts with retrospective provenance. Retrospective provenance is the information on what has happened in a particular experiment [7]. Prospective and retrospective provenance are neither mutually exclusive nor mutually dependent. Some systems accommodate one or the other, and some accommodate both.

In 2005, The EU Provenance Project developed the Open Provenance Model (OPM) to create an open-source, common data model. It supports a “technology-agnostic” system general enough to be useful to everyone [5]. Its underlying storage structure is irrelevant, and it never attempts to define or restrict storage mechanisms, syntax, or querying [15, 5]. The OPM is limited by the fact that it stores only retrospective provenance. Further, it has never been widely adopted.

Provenance systems do not exist in a vacuum. One system can dramatically influence another. This paper is an attempt to view provenance systems in their appropriate context and to identify areas for continued improvement.

The paper is organized as follows: Section 2 describes several actual provenance systems. Section 3 discusses observations that can be made from the discussion of provenance systems. Section 4 contains concluding thoughts.

2. PROVENANCE SYSTEMS
This section describes several provenance systems in semi-chronological order. Obviously, system development takes time, often years, so a strict chronology based on release dates can be misleading. Thus, a few systems are listed anachronistically for clarity.

2.1 FITS
Among the first provenance systems is the Flexible Image Transport System (FITS). FITS was developed by NASA and released in the early 1980s. FITS uses metadata stored in file headers to maintain provenance data about a particular file [23]. It has no centralized data model per se. As such, querying FITS metadata can be extremely difficult [23].

NASA wanted an archive format that was portable under the assumption that the hardware and software that wrote the data will not be available when the data is read [10].

2.2 Trio
Trio, the first paper for which was published in 2005, is a system developed specifically for the relational database environment. It grew out of a desire to expand the capabilities of data warehouses to accommodate provenance information. There were other systems that handled data provenance over specialty databases, but Trio was an attempt to implement data provenance within a general relational environment [4]. Furthermore, Trio is an attempt to improve over the previous database-specific provenance systems, as it ensures the graceful handling, storing, and querying of inexact data [4, 25].

Viewing the provenance of a data item involves utilizing query inversion to progress backwards through the steps used to create the tuple of interest [4, 25]. The actual Trio data model is a layer of abstraction on top of a relational database. This data can be queried using The Trio Query Language (TriQL). TriQL is an extension of SQL specifically designed to query both data provenance and accuracy [25].

2.3 Chimera/VDS
Chimera is a provenance system introduced in 2002 within the GriPhyN project. Chimera is a prototype system for the Virtual Data System (VDS), used to maintain provenance and other information in a grid environment. The finalized
VDS was released in 2003. The Chimera/VDS model consists of two basic components: the virtual data catalog (VDC) and the virtual data language interpreter. The VDC is the mechanism that implements Chimera/VDS’s virtual data schema. The schema is system-defined, and user data is added as needed. The virtual data language interpreter executes statements written in Chimera/VDS’s virtual data language (VDL). VDL is a specialized language, which, like SQL, is used to both populate and query the Chimera/VDS database [6].

2.4 CMCS

The Collaboratory for Multi-Scale Chemical Science system (CMCS) is a provenance system developed in 2003 specifically to work in the field of chemical science. CMCS uses Scientific Annotation Middleware (SAM) to manage the data and metadata [19]. Thus, SAM is charged with maintaining storage. Yet, SAM is storage-agnostic [17]. It uses the Distributed Authoring and Versioning (DAV) protocol to communicate with the storage systems [17, 20].

Data is entered into the system via the DAV protocol or manually using the CMCS web portal [18, 19]. The web portal allows for direct entry of legacy data. The DAV protocol also allows for applications to enter their own metadata automatically [18]. Unfortunately, this requires the applications be aware of the CMCS system and willing to use it. If they are not, the metadata they produce can be manually entered via the portal.

2.5 Karma

Karma is a workflow system released in 2005 and is part of the LEAD Cyberinfrastructure. It collects provenance information by encapsulating processes in its custom Application Factory Toolkit, a generic web service wrapper [22]. Karma distinguishes between data and process provenance. Data provenance describes how data came to be. Process provenance is the execution details of an instance of a process [22, 24]. It stores provenance data in a standard relational database [22].

Karma’s query capabilities are not as straightforward as other systems. Attempting to answer the queries in the First Provenance Challenge, Karma uses three different levels of processing. Two of the nine queries were answered by single calls to the Karma server API. Two were answered by multiple calls to the Karma API with some client-side processing. Four of the queries had to use SQL to query the underlying Karma database. One query was unsupported [22].

2.6 VisTrails

VisTrails is an open-source workflow system. It stores workflows as dataflows, workflows where the nodes represent data and the edges represent transformations between data. It contains a complex GUI used to build and query provenance information. Dataflow specifications are stored as XML and are stored separately from the execution of the dataflows [2]. Thus, VisTrails stores both prospective and retrospective provenance. Specifications are the prospective provenance, and execution details are retrospective.

The VisTrails query mechanism is Query by Example (QBE) and is done through the GUI [21]. Dataflow specifications can be queried just as easily as provenance information [2].

2.7 Perm

Like Trio, the Perm provenance management system is an extension of the relational database model. It attempts to correct issues inherent to Trio. For example, in Trio, the result of a provenance query is a relation that is incompatible with relational algebra, meaning further computation on the result must be handled by Trio itself and does not benefit from any optimization of the DBMS [8]. Further, when Perm was developed, Trio contained only limited support of SQL statements. Perm’s developers wished to expand this [8].

Perm provenance queries are issued using an SQL extension called SQL-PLE. The original SQL language is preserved in full. SQL-PLE simply adds some keywords that are useful in querying provenance information [8].

2.8 Non-Scientific Systems

Microsoft, Yahoo!, and Apple each created a workflow system aimed at the tech-savvy user. Microsoft’s system is the Windows Workflow Foundation1, released in 2006. Yahoo! created Yahoo! Pipes2 in 2007. Apple created Automator3 as part of MacOS X v10.4, which was released in 2005. These systems all eschew scientific functionality, in favor of aiding the user in creating workflows that are applicable to his or her daily life. The systems are all free, but Windows Workflow Foundation and Automator work only on Windows and MacOS respectively.

2.9 Other Systems

There are several systems which could not be included here. Vesta [11] was designed to compensate for perceived deficiencies in existing software configuration systems. Pegasus is a system for both creating workflows and automatically mapping them to grid computing resources [13]. Kepler is an open-source workflow system that was an attempt to improve upon the Open Provenance Model by extending it to handle workflows [1]. Taverna is a workflow system within the myGrid project that attempts to improve Trio’s query inversion to work in places it previously had not [14]. PASS is a provenance application that employs a custom file system, stores provenance data as a directed acyclic graph, and employs new query (nq), a custom query language, as one of its query mechanisms [12].

3. OBSERVATIONS

Some observations can be made from the descriptions of provenance systems above. This section discusses some of those observations and points out areas for continued study.

3.1 Perspective

Individual provenance systems grew out of one of two origins. Some were generated in direct response to a need voiced by a particular area of study. Others were created as academic or commercial exercises and were not designed to service any particular need. Examples of need-based systems include FITS, Vesta, and CMCS. Examples of a generic system include VisTrails, Taverna, Kepler, Trio, Perm, and PASS.

2http://pipes.yahoo.com/pipes/
3http://support.apple.com/kb/HT2488
Need-based systems are generally useful in one specific area but have limited usability outside that area. FITS, for example, has the portability that NASA wants [10]. On the other hand, querying is a function of nearly every other provenance system. FITS lacks a defined query mechanism [23].

3.2 Storage

The actual storage mechanism used by different systems varies. Some systems like Trio and Perm are designed for specific environments and utilize those environments extensively. VDS and the systems based on it (Chimera, and Pegasusus) can be represented as a relational database, an object-oriented database, an XML repository, or a hierarchical directory [6]. Trio and Perm, designed for relational databases, store data in database tables [9, 8, 4, 25]. Other systems like Pegasusus and Karma, though not designed to store provenance only about a relational database, still use relational databases to store data [13, 22]. These systems avail themselves of the efficiency of relational databases. Depending on the abstract data model, it is possible for querying to take advantage of SQL as well, but that will be discussed later.

Some systems are storage-agnostic like CMCS and the OPM [17, 5]. Instead, they concentrate on message passing both within the systems and between the system and other programs. The obvious benefit is that it allows the user to specify his own storage device. This, of course, requires a great deal more attention to that end. While provenance systems like Trio fit neatly on top of an existing database, with storage-agnostic systems, much more attention must be paid to the physical storage device, as an administrator must both install the system and ensure proper communication between the provenance system and its data store.

The middle ground between a pre-defined data store and a storage-agnostic system is a program like VDS. The virtual data schema can be implemented atop several different storage devices. However, the virtual data model is abstracted to a level above the physical storage such that most users will not have to consider physical storage. A virtual data language exists to provide a consistent query language across several types of physical stores, but it requires users learn the language [6].

3.3 Data Model

Unfortunately, not all systems go into great detail about their data models. Trio works on a layer of abstraction above the relational database model, but it is still based on the relational model [25]. Perm sacrifices some of Trio’s features like the recording and querying of accuracy data in favor of a model tied more closely to the relational model [9, 8].

Several systems conceptualize their data as a directed acyclic graph. Taverna, PASS, and VDS all utilize DAGs [14, 16, 3]. There is a division on how the DAG is constructed. Workflows usually consist of both data (i.e., files, tuples, etc.) and the processes used to create and modify data. As such, DAGs are traditionally represented as either data-centric (i.e., dataflows) or process-centric. Dataflows store data as nodes and processes as edges. Process-centric graphs are the reverse. Graphs generally emphasize nodes over edges, and it is easier conceptually and often physically to store complex data on nodes as opposed to edges. This forces the user to commit to one or the other.

3.4 Querying

Storing data is useless unless it can be viewed or used internally. Thus, querying is extremely important. Virtually all domains require some sort of querying. Some queries will be issued by technical users, and some will not. This is domain-specific and is a vital consideration when evaluating the usefulness of a generic provenance system.

Some systems like Kepler and PASS provide a GUI to query provenance data [12]. This can provide a user with a consistent and intuitive interface through which to query what can easily become immensely complex data.

Unfortunately, most provenance systems employ some sort of custom query mechanism, usually a query language. VDS (and systems based on it), Perm, PASS, and Trio all use custom languages. VDS uses the virtual data language [6, 3]. Perm uses SQL-PLE, an extension to SQL [9]. PASS contains a command-line supplement for its GUI that uses a language called new query (nq) [12]. Trio uses TriQL, an extension of SQL [25].

4. CONCLUSIONS

Despite efforts by the EU Provenance Project, true standards in data provenance have yet to be adopted. There are many different opinions on open research issues, and the development of competing systems led to the creation of conflicting standards. Some systems were created to fulfill a specific need, and those systems, though they thoroughly address the need, often have little real application outside of their home niche. Others, though often funded by a specific grant or project, were developed as general-purpose systems. The general-purpose systems are so general that they are often extremely complex and difficult to learn. Some systems are storage-agnostic, and some are closely tied to their storage mechanism. Storage-dependent systems can take advantage of the efficiency of the underlying storage device, but storage-agnostic systems are more general. Few systems give extensive details about their data models. Most that do, store utilize a DAG. Some systems use DAGs whose nodes are data, and some use DAGs whose nodes are processes. Attempting to query the data also lacks standards. Most systems employ some custom query language, which requires the user learn the new language. The only working solution in this area is to use a GUI and/or QBE.

Some of the rougher edges in data provenance are being polished by non-scientific data provenance systems. Microsoft, Apple, and Yahoo! do not have the luxury of assuming their users are computer programmers, which requires stricter standards be adopted by each system. Even if the companies do not share standards, only three competing standards would certainly be an improvement over the current state of affairs.

5. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under award number EPS-0903787. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the National Science Foundation.

6. REFERENCES


