A Survey of Data Provenance Techniques

Yogesh L. Simmhan, Beth Plale, Dennis Gannon

Computer Science Department, Indiana University, Bloomington IN 47405

{ysimmhan, plale, gannon}@cs.indiana.edu

Technical Report IUB-CS-TR618

Abstract

Data management is growing in complexity as large-scale applications take advantage of the loosely coupled resources brought together by grid middleware and by abundant storage capacity. Metadata describing the data products used in and generated by these applications is essential to disambiguate the data and enable reuse. Data provenance, one kind of metadata, pertains to the derivation history of a data product starting from its original sources.

The provenance of data products generated by complex transformations such as workflows is of considerable value to scientists. From it, one can ascertain the quality of the data based on its ancestral data and derivations, track back sources of errors, allow automated re-enactment of derivations to update a data, and provide attribution of data sources. Provenance is also essential to the business domain where it can be used to drill down to the source of data in a data warehouse, track the creation of intellectual property, and provide an audit trail for regulatory purposes.

In this paper we create a taxonomy of data provenance techniques, and apply the classification to current research efforts in the field. The main aspect of our taxonomy categorizes provenance systems based on why they record provenance, what they describe, how they represent and store provenance, and ways to disseminate it. Our synthesis can help those building scientific and business metadata-management systems to understand existing provenance system designs. The survey culminates with an identification of open research problems in the field.

1 Introduction

The growing number and size of computational and data resources coupled with uniform access mechanisms provided by a common Grid middleware stack is allowing scientists to perform advanced scientific tasks in collaboratory environments. Large collaboratory scientific projects such as the Large Hadron Collider [1] and Sloan Digital Sky Survey (SDSS) [2] generate terabytes of data whose complexity is managed by data grids. This data deluge mandates the need for rich and descriptive metadata to accompany the data in order to understand it and reuse it across partner organizations. Business users too are having to work with data from third-parties and from across the enterprise that are aggregated within a data warehouse. Dash-boarding tools that help analysts with forecasting and trend prediction operate on these data silos and it is essential for these data mining tasks to have metadata describing the data properties [3]. Provenance is one kind of metadata which tracks the steps by which the data was derived and can provide significant value addition in such data intensive scenarios.

Provenance (also referred to as lineage, pedigree, parentage, genealogy, and filiation) can been described in various terms depending on where it is being applied. Buneman et al [4] define data provenance in the context of database systems as the description of the origins of data and the process by which it arrived at the database. Lanter [5] refers to lineage of derived products in geographic information systems (GIS) as information that describes materials and transformations applied to derive the data. Provenance can be associated not just with data products, but with the process(es) that enabled the creation of the data as well. Greenwood et al [6] expand Lanter's definition of provenance and view it as metadata recording the process of experiment workflows, annotations, and notes about experiments. For the purposes of this paper, we define data provenance to be information that helps determine the derivation history of a data product, starting from its original sources. We use the term data product or dataset to refer to data in any form, such as files, tables, and virtual collections. The two important features of the provenance of a data

product are the ancestral data product(s) from which this data product evolved, and the process of transformation of these ancestral data product(s), possibly through workflows, that helped derive this data product.

In this paper, we provide a detailed view of current data provenance research in the scientific and business domains. Based on a survey of the literature on provenance, we develop a taxonomy of provenance techniques to assist us in analyzing and comparing efforts in this sphere at the conceptual level. Subsequently, we apply this systematic to contrast nine key provenance systems and concepts that have been proposed or are under active research. We present some early publications that have laid the foundation for provenance and provide a historical context for approaching the problem. But a significant portion of the survey is devoted to current challenges that are driving research in this field. Our synthesis can help those managing scientific and business metadata to understand existing provenance system designs and incorporate provenance into their data processing framework. We conclude with a discussion of open research problems and challenges to managing provenance.

While data provenance has been gaining interest in the recent past due to unique desiderata introduced by distributed data in Grids, few sources are available in the literature that compare across approaches. Bose et al [7] survey lineage retrieval systems, workflow systems, and collaborative environments, with the goal of proposing a meta-model for a systems architecture for lineage retrieval. Our taxonomy based on usage, subject, representation, storage, and dissemination more fully captures the unique characteristics of these systems. Miles et al [8] study use cases for recording provenance in e-science experiments for the purposes of defining the technical requirements for a provenance architecture. We prescribe no particular model but instead discuss extant models for lineage management that can guide future provenance management systems. Two recent workshops on data provenance, derivation, and annotation [9, 10] brought forth positions papers on current research in this subject. Some online bibliographies on provenance also exist [11, 12].

This survey is structured as follows. In section 2, we provide background information that motivate the need for provenance, and discuss data processing frameworks in which provenance plays a role. In section 3, we present the provenance taxonomy we developed drawn from the papers we surveyed. We apply this taxonomy in section 4 to compare and contrast nine projects that taken together give a comprehensive overview of research in this field. In section 5, we identify open research problems and challenge in the field and conclude with a summary of our survey in section 6.

2 Background

2.1 Motivating Domains for Provenance

Provenance finds its use in academic and research organizations, as well as in business establishments, but the data organization is different in both these domains. In the following sub-sections, we motivate the necessity of provenance for these application domains, and bring out the differences in the way provenance is collected and used in them.

2.1.1 Scientific Domain

Data used in the scientific field can be ad hoc and driven by individual researchers or small scientific communities. However, the scientific field is moving towards more collaborative research and organizational boundaries are beginning to disappear in the face of dynamically created Virtual Organizations (VO) [13]. Sharing data and metadata across organizations is essential in such a collaborative environment, leading to a convergence on common schemes to ensure compatibility. Despite this, scientists consistently deal with greater heterogeneity in data and metadata than business users. Issues of trust, quality, and copyright of data are significant when using third-party data in such a loosely connected network, and provenance metadata can address some of these concerns.

Scientific domains use provenance in different forms and for various purposes. Scientific publications are a common form of representing the provenance of experimental data and results. Increasingly, Digital

Object Identifiers (DOIs) [14] are used to cite these data used in experiments so that the papers can relate the experimental process and analysis – which form the data's lineage – to the actual data used and produced. Some scientific fields find it valuable to go beyond this and store lineage information in a machine accessible and understandable form so that the experimental methods may be validated more easily.

GIS standards suggest that metadata about quality of datasets should include a description of the lineage of the data product [15]. The Spatial Data Transfer Standard (SDTS) [16] specifies that lineage should contain, among others, a description of the source material from which the data was derived, the transformations used to derive it, references to the control information (such as permanent geodetic reference points [17]) used, and mathematical transformations of coordinates used. Recording lineage helps the user of the data product to decide if the data meets the requirements of their application [18].

Materials engineers choose materials for the design of critical components, such as for an airplane, based on the statistical analysis of test data provided on the material [19]. It is critical to establish the pedigree of this data since bad data can have disastrous consequences during manufacturing and in the product's performance. It can also help to locate sources of faulty components in case a system fails at a later stage.

Presence of transformation history promotes the sharing of biological and biomedical data in life sciences research [20, 21]. Provenance of the data provides such information and, analogous to citations in publications, serves to acknowledge its author. Provenance gives a context in which to use the data, and allows automated validation and revision of derived data when the base data is updated. Knowledge of provenance is also relevant from the perspective of regulatory mechanisms to protect intellectual property. For example, the United Nations Environment Program is proposing a provenance certificate for genetically modified biological resources to protect the genetic resources of countries from being exploited [22].

Astronomical sky surveys, such as the Palomar Digital Sky Survey (DPODSS) [23] and Sloan Digital Sky Survey (SDSS) project [24] make uniform astronomical data available publicly through online digital archives. This allows astronomers to run their mining algorithms and visualization tools on these federated datasets and publish their validated results [25]. In such a collective effort that uses data integrated from third-party sources, recording the provenance of the results can help astronomers estimate the trust placed in them [24].

With a growing number of datasets available in the public domain beyond the confines of a single organization, it has become increasingly important to determine the veracity and quality of these datasets. The above examples support this view. It does not suffice to have an abstract notion that a dataset is from a reliable source, but this has to be backed by a detailed history of the data that will allow the user to apply their own metrics to determine if the data is acceptable. Lineage information about the data can provide semantic meaning to it and help integrate it within the local data processing framework of the scientist. Identifying the source of the data also helps to credit the creator of the data.

2.1.2 Business Domain

Business users traditionally work with an organized data schema, where the structure and semantics of the data in use is shared across the corporation [26]. Even business-to-business (B2B) data communication relies on clearly established schemes for data interchange [27] and usually with trusted partners. These factors contrast with the scientific domain and make it easier for businesses to trust the data source. Yet, a large proportion of businesses deal with bad quality data, and this is accentuated when they are aggregated from different parts of the enterprise into a data warehouse [28]. Sources of bad data need to be identified and corrected to maintain the data quality and avoid costly errors in business forecasting.

Data warehouses provide an integrated view across historical data from multiple sources while retaining the depth of the data and summarized information on it [29]. Analysts use business intelligence and business analytics tools to mine this data and assist in decision-making. A process of extracting,

cleansing, and transformation steps are applied to data from multiple operational databases and external sources to identity relevant information and normalize their representation before being loaded into the warehouse [30, 31]. This consolidated view in the data warehouse is updated frequently to reflect the changes in the source databases, and maintain its consistency and timeliness [32].

In a warehouse environment, lineage information is used to trace the data in the warehouse view back to the source from which it was generated [33]. This can be challenging given that the warehouse data is built upon layers of data views, with one layer derived from data in layers below it [34]. Lineage aids the data analyst in "drilling through" to the source of a particular data item in the view and explore additional characteristics of the data source not available in the data warehouse [35, 36]. Additionally, it helps to trace faulty data back to the source of errors in order to apply the relevant correction to it [37]. From a database theory perspective, this operation is similar to the view deletion problem [38] which involves locating the source data which should be modify in order to delete data appearing in a view.

2.2 Data Processing Architectures

Provenance information for a data product is centered on two concepts: the ancestral data products and the transformations that they underwent to produce that data. It is convenient to conceptualize the data products and transformations in the form of a graph (more specifically, a directed acyclic graph or DAG), with nodes representing the data products and the edges the transformation processes[1]. Data processing architecture refers to the means by which these processes execute, consume data products, and bring about the transformation of the data. Bose et al [7] use data processing system as a means to categorize lineage retrieval systems. We consider the architecture used for data processing as more of an implementation artifact independent of our provenance taxonomy, but do describe some of the distinctive ways in which provenance is collected in them.

2.2.1 Service-oriented Architecture

Computational Grids are powerful platforms that help teams of researchers to collaborate and solve challenging problems on widely distributed resources in a relatively seamless manner [39]. Grid frameworks provide service and component based programming models for building applications that run on the Grid [40]. Web services and Grid computing together provide a service-oriented platform for data processing that is popular with the scientific community.

A service-oriented architecture usually allows a transformation graph, such as a DAG described earlier, to be specified in the form of a workflow document written in languages like WSFL [41] or BPEL [42]. The transformation processes are modeled as web or Grid services and the data products are the inputs and outputs to these services (**Figure 1**). The provenance for datasets involved in the workflow can be determined by tracing the execution of the workflow and identifying the input and output data products to each service by means of logical or physical IDs. These traces can be automatically generated by the workflow engine and later annotated by the users to provide additional metadata on the data product. Static information available about the workflow can be combined with the runtime details to form the provenance, though this may not be possible in the case of dynamic workflows that are programmed to adapt with current external conditions [43]. If the workflow trace is not collected by the workflow engine, the distributed nature of services puts the onus on each service provider and client to generate a log of their invocation that is aggregated to form the provenance trace for the workflow. This requires agreement on some standard way to record and share the provenance [44]. The lineage techniques of Chimera [45], myGrid [46], CMCS [47], and PASOA [44] presented in this survey use a service-oriented architecture.

[1] Alternatively, we can also conceptualize the data products as vertices in the DAG with processes being the edges. The two are inverse representations of the same information.

2.2.2 Database Architecture

In a database architecture, update queries and functions form the data processing component that transform the data. A data product in a relational database can be a view, a table, a tuple, an attribute, or a data item of even finer granularity in the database. It can also be a pointer to an external data resource such as a file [48]. In this architecture, a data product's lineage can be traced back through a series of functions and update query requests.

Data warehouses are an archetype for a data processing system using databases and benefit from tracking lineage as discussed in section 2.1.2. Warehouses import data through extraction, cleansing, and transformation steps modeled as queries over multiple data sources. In addition to the typical relational operators like select, project, and join, these queries can invoke user-defined functions that are implemented as stored procedure calls. From this rich capability, one can construct intricate dataflow graphs that are executed by the database as part of the query (**Figure 2**). This gives a capability analogous to workflows in the service-oriented architecture. The lineage of query results in a database can be denoted using annotations on attributes in the databases [49]. These lineage annotations encode information about the data source and the query that created them [50]. Databases can also use query inversion and function inversion techniques to trace the lineage from a data item back to its source [4, 51]. These annotation and inversion techniques are discussed further in section 3.3.1.

Not all queries and user-defined functions are capable of being inverted, and this restricts the usefulness and accuracy of the provenance. Databases are also more rigid when it comes to adding ad hoc annotations to data. There may also be problems when tracing lineage of externally linked data sources, such as files, that are processed outside the database, and the lineage chain may disconnect at this boundary [48]. Federated databases pose additional challenges for tracking provenance due to the source transparency they provide over heterogeneous data [52]. Lineage for these heterogeneous data source will be disparate and there should be a way of making lineage for them also transparent. The lineage techniques of Tioga [53], P. Buneman [4], and Trio [54] presented in this survey use a database architecture.

2.2.3 Other Data Processing Architectures

In addition to service-oriented and database architectures, command processing architectures and script-based architectures are also used, though less frequently [7]. Users in a command-processing model interact with the data processing system through commands entered in a shell interface or batch executed from files. These commands can perform transformations on data products managed by the data management subsystem. The shell interface usually has a provision to log the commands that are executed along with their associated inputs and outputs to enable debugging [55]. Lineage information can be collected from these log files analogous to the workflow trace in the workflow architecture, with additional metadata explicitly provided by the user. This provenance metadata may be stored in the data management subsystem along with the data products. Command processing systems are not in common use but their relevance to this survey lies in that early investigations into data lineage were performed by D. P. Lanter [55] on such command systems for use in GIS.

Scripting architecture is more contemporary and is popularly used by the scientific community to execute their processing applications. They provide a modular programming model, and are easy to learn and use. Scripting environments, such as Matlab, Jython, and Perl, provide powerful libraries to enable advanced tasks such as interacting with databases [56], and composing and invoking web services workflows [57]. Inputs and outputs to scripts are through command-line parameters or loaded from an input list file containing the arguments to the script. A "workflow" script or a separate workflow application can invoke various scripts in succession to perform complex scientific experiments. Data flow between scripts is accomplished by passing references to datasets files as part of the script parameter or by embedding the dataset location within the code. Scripts use internal commands or external libraries to log their execution, and this may be extended to generate lineage information about data products used and created within the

scripting framework. Special libraries can be used to construct provenance metadata [58] and the logs from individual scripts in the workflow amalgamated to construct the lineage for the data products participating in the workflow. The onus is on the script-writer to provide the metadata about its activity that will go towards determining the provenance of the datasets. Application frameworks that retain the flexibility of scripting while providing automated recording of provenance during their execution would ease the burden on the script-writer. The lineage techniques of ESSW [58] presented in this survey use a service-oriented architecture.

3 Taxonomy of Provenance Techniques

Different approaches have been taken to support data provenance requirements for individual domains. In this section, we present a taxonomy of these techniques from a conceptual level with brief discussions on their pros and cons. A summary of the taxonomy is given in **Figure 3**. Each of the five main headings is discussed in turn.

3.1 Application of Provenance

Provenance systems may be constructed to support a number of uses [59, 60], and Goble [61] summarizes several applications of provenance information as follows:

- *Data Quality*: Lineage can be used to estimate data quality and data reliability based on the source data and transformations [20]. It can also provide proof statements on data derivation [62].
- *Audit Trail*: Provenance can be used to trace the audit trail of data [63], determine resource usage [6], and detect errors in data generation [37].
- *Replication Recipes*: Detailed provenance information can allow repetition of data derivation, help maintain its currency [63], and be a recipe for replication [64].
- *Attribution*: Pedigree can establish the copyright and ownership of data, enable its citation [20], and determine liability in case of erroneous data.
- *Informational*: A generic use of lineage is to query based on lineage metadata for data discovery. It can also be browsed to provide a context to interpret data.

We expand on and more clearly define these applications below for the purposes of our classification.

3.1.1 Data Quality

Provenance about a dataset enables its user to evaluate its quality for their application. Data quality of source data is important since errors introduced by faulty data tend to inflate as they propagate to data derived from them [65]. This issue is even more acute when using data available off public archives, as is common in genomics [20, 66]. The level of detail included in the provenance determines the extent to which the quality of the data can be estimated. Rudimentary lineage metadata about the data, such as the transformation applied to create it or the source of its parent data, can assist the data user in establishing the authenticity of the data and avoid spurious sources. If certified semantic knowledge of the pedigree is available, it is possible to automatically evaluate it based on quality metrics that are defined and provide a quality score using modeling techniques [65, 67, 68].

3.1.2 Audit Trail

Provenance can serve as a means to audit the data and the process by which it was produced. Such information can be important when establishing patents on drug discovery or for accounting purposes for businesses [8, 69]. This can also be used to optimize the derivation process [45], and collect statistics to account for resource usage [6]. Lineage in the form of a runtime trace of execution can help in verifying if any exceptions took place in data creation. A recurrent use of provenance is to backtrack and locate the source data or process that is the cause of errors found in derived data and apply relevant corrections [37].

3.1.3 Replication Recipes

Provenance information includes the steps used to derive a particular dataset and can be thought of as a recipe for creating that data [64]. If the provenance contains sufficient detail on the operations, data sources, and parameters, it may be possible to repeat the derivation. Repeatability entails the availability of similar resources as was available when the original data was created. The derivation may be repeated to maintain the currency of derived data when then source data changes or if the processing modules were modified. Such a re-enactment can be controlled to repeat just those sections affected by the change in base data or process [8]. This is similar to the view maintenance problem is databases where database views derived from underlying source tables and views need to be updated when the source tuples change [4, 70]. It may be possible and cost effective to use provenance as a means of replicating the data instead of transporting it or storing it. For the derived data to be physically identical, several dependencies may have to be met such as access to the same source data, processes, and processing environment. In some cases like a stochastic experimental run, such a byte-for-byte replication may be impossible but a semantically equivalent data product could be generated [64]. Such properties can be extended to compare two datasets by just comparing their lineage, rolling back changes, and reverse engineering the data.

3.1.4 Attribution

Pedigree can help ascertain the ownership of the source data used to generate a certain data. Users can look up the derivation tree to see the creators of the source data and verify its copyright [71]. Similarly, creators of intellectual property can look down the lineage chain to see who are using data that they created. Citations are an important part of publication in science and lineage acts as one form of citation when publishing scientific datasets. It can also be used as a means of assigning liability in case of errors in the dataset [60].

3.1.5 Informational

A generic use of provenance is as a metadata description that can be the basis for datasets discovery, say by searching based on source data or a processing step used to generate them. Such queries can locate data of interest and avoid duplication of effort if the same derivation has already been performed. Annotations provided along with provenance can help to interpret the data in the context it was intended, especially for archived data that are used long after they are generated [72]. This helps to unambiguously assimilate data into the user's application domain. Provenance can also be browsed as a derivation tree or in other graphical forms, and act as a starting point for exploring other metadata about the data and processes.

3.2 Subject of Provenance

Provenance information can be collected about different resources present in the data processing system and at various levels of detail. It may be more applicable to collect provenance about certain types of data products than on others, based on the importance of the data or the cost of collecting provenance for it [60]. We classify provenance systems on the basis of the subject that the provenance describes and its granularity.

3.2.1 Data vs. Process Oriented Provenance

The provenance techniques we surveyed focus on data, but this data lineage can either be available explicitly or deduced indirectly. In an explicit model, which we term a *data-oriented* model, lineage metadata is specifically gathered about the data product. One can delineate the provenance metadata about the data product from metadata concerning other resources. For example, a DAG of transformation steps that derived a data product can be ascribed as its lineage, and this DAG is precisely associated with just that dataset without having to derive it from some external source. This contrasts to a *process-oriented*, or indirect, model where the deriving processes are the primary entities for which provenance is collected, and the data provenance is determined by inspecting the inputs and outputs of these processes. For instance, myGrid [73] collects provenance information in the form of a workflow trace centerd around the services in the workflow, and this is used to derive the data provenance of datasets generated

during the workflow's execution [74]. Depending on the application context in which provenance is captured, either of these methods may be used [75]. There may not even be any clear distinction on whether provenance is recorded about the process or data, and may rather have the ability to extract either forms of provenance.

3.2.2 Granularity of provenance

The usefulness of provenance to a certain domain is linked to the level of granularity at which it is collected. The domain requirements can call for provenance to be recorded on attributes or tuples in a database that represent individual pixels [53] or array elements [76]. Alternatively, files in a collection generated by the same experimental run can have similar provenance and it may suffice to store a statistical aggregate of the provenance for all files in the collection [77]. Data products that are a subset of a parent dataset [78] may share their provenance with the parent or sibling and yet be different as a whole. Some domains require provenance to be stored at multiple levels of granularity and this calls for a flexible approach to be taken by the provenance system. The use of abstract datasets [64, 74], that refer to data irrespective of granularity or format, are increasing in use and provide such flexibility. This makes the provenance collection independent of the granularity or representation of the dataset. The cost of collecting and representing provenance can be inverse to its granularity and this will play a role in the granularity needed.

3.3 Representation of Provenance

Different techniques can be used to denote provenance information, some of which are dependent on the underlying data processing system. The manner in which provenance is represented has implications on the costs for recording it and the richness of its usage. We classify provenance systems according to the schemes used for representing provenance, the contents of their lineage, and the semantic details incorporated.

3.3.1 Scheme for Storing Provenance

The two major approaches to representing provenance information use either annotations or inversion. In the former, metadata comprising of the derivation history of a data product is collected as *annotations* and descriptions about sources data and processes. This is an *eager* form [50] of representation in that provenance is pre-computed and readily usable as metadata. Alternatively, the *inversion method* uses the property by which some derivations can be inverted to find the input data supplied to derive the output data [38, 51]. Examples of such derivations include queries and user-defined functions in databases that can be inverted automatically or by explicit functions. The derivation queries are used to create an inverse query that operates on the output data and auxiliary data to identify the source data. Similarly, inverse functions defined for user-defined functions take the results of the user-defined functions as parameter and return their input data. The inversion technique is related to the view updation and view deletion problem in databases where in order to update or delete a view, we need to identify the source tables that need to be modified.

While formal techniques are available to determine the inverse query of a relational query, they are restricted to a certain class of relational queries and are not universally applicable. Similarly, not all user-defined functions have inverse functions. The advantage of the inversion method is its (arguably) compact representation of the provenance, compared to annotation method, whereby an entire class of derived data can be represented concisely using a single inverse query or function. But annotations give more flexibility in the richness of provenance metadata and the provenance need not be computed "just-in-time" like in the inversion method.

3.3.2 Contents of Provenance

Lineage information provided by the inversion method is sparse and limited to the derivation history of the data. It stores just the query or process that created the derived data ("why" the data was created) and they serve to identify just the source data that created the derived data ("where" they originated from) [4,

54]. Annotations, on the other hand, can be richer and, in addition to the derivation history, often including the parameters passed to the derivation processes, the versions of the workflows that will enable reproduction of the data, or even related publication references. These may be sufficient to repeat the derivation process or reproduce the derived data. It is a moot point on where the boundary between provenance information and generic metadata lies. In some cases, there is little to distinguish the two and provenance is subsumed into the general metadata infrastructure.

3.3.3 Provenance Format: Syntactic Structure and Semantic Information

There is no metadata standard for lineage representation across disciplines and, due to their diverse needs, it is a challenge for a suitable one to evolve [79]. Many current provenance systems that use annotations have adopted XML for representing the lineage information [44, 46, 47, 80]. The benefits of this are apparent given that many of them use a service-based architecture where XML is the primary format for message exchange. The format for representing lineage in the inversion method is less relevant since it is likely to be dependent on the query format (such as SQL in relational databases) or the user-defined function used to process the data.

Some of the annotations also capture semantic information within provenance using domain ontologies in languages like RDF and OWL [46, 47]. Ontologies clearly establish the concepts and relationships used in the provenance metadata and provide portable contextual information. Encoding of semantic knowledge allows an enhanced use of provenance, and helps to reason about and provide proof statements about the lineage of the data [62, 81]. The flipside to this is the effort involved in defining domain ontologies and giving a semantic description to each data and process in the system.

3.4 Provenance Storage

Provenance information can grow to be larger than the data it describes if the data is fine-grained and provenance information rich. So the manner in which the provenance metadata is stored is important to its scalability. Provenance can be tightly coupled to the data it describes and located in the same data storage system or even be embedded within the data file. Some data formats such as Flexible Image Transport System (FITS) [82] and Spatial Data Transfer Standard (SDTS) [16] allow the use of metadata headers, and lineage can be one such header entry. Such approaches ease maintaining the integrity of provenance, but make it harder to publish and search just the provenance. Provenance can also be stored separately with other metadata or simply by itself. In maintaining provenance, we should consider if it is immutable, whether it should be versioned when the data is modified, or if it should be updated to reflect the modified state of its predecessors [61]. If the data's version changes due to an update to its deriving process or source data, the provenance information for the new version of the data can overlap significantly with the previous version, providing possiblities to optimize the storage [83]. The provenance collection and storage repository also determines the trust one places in the provenance and if any provenance mediation services are required [8].

3.4.1 Scalability

Provenance systems can scale with the number of datasets, their granularity, the depth of the lineage (generations of ancestral data), their geographical distribution, and the user base. The number of datasets for which provenance is to be recorded is domain dependent and related to the granularity of the dataset, and the number of datasets increase as the granularity decreases. As the stages required to derive a dataset grows, the depth of its lineage correspondingly increases. It may be required to recursively inspect the provenance of each of its ancestral data in order to assemble the complete provenance for a dataset. The cost for gathering these increases exponentially with the depth of provenance[1], and is accentuated if the

[1] We can consider the provenance for a dataset as being derived by an inverted DAG where the dataset is at the root and the deriving datasets and processes (its ancestors) are its child nodes (refer to **Figure 1)**. As the depth of this graph increases, the number of child nodes (ancestral data) increases exponentially.

provenance of the ancestral data sources is geographically distributed across multiple repositories [64, 84].

The inversion method used to represent provenance scales well with the number of datasets and their granularity since an inverse function or query identifies the provenance for an entire class of data [53]. The amount of auxiliary source data required to find the provenance by this method is a crucial factor in ensuring its scalability [38, 51]. The inversion method is not well suited if the auxiliary data is geographically distributed as the inverse query needs to fetch them before it is executed. Database optimization techniques can make the inverse query efficient and the computed provenance can be cached to avoid repeating the query, provided provenance is immutable.

Annotations may not scale well for fine-grained data as the complete provenance for the data may outsize the storage space required for the data itself [53]. Even for coarse-grained data, the size of annotation increases exponentially as the depth of lineage increases. However, one can reduce storage needs in the annotation method by recording just the immediately preceding transformation step that creates the data and recursively inspect the provenance information of those ancestors for the complete history.

3.4.2 Provenance Overhead

Management of provenance incurs costs for its collection and for its storage. The size of provenance information may exceed that of data's and the storage cost of rich provenance can be significant. Less frequently used provenance information can be archived to reduce storage overhead [60], or a demand-supply model based on usefulness can retain provenance for those frequently used. Such a model can promote better provenance description to be stored with the data since datasets with rich provenance is likely to be used frequently and hence persisted.

If provenance depends on users manually adding annotations instead of automatically collecting it, the burden on the user may prevent complete provenance from being collected. While it may not be possible to automate the collection of certain forms of provenance (such as inverses for a user-defined function), instrumented data management middleware should be able to generate provenance during the data creation process. Automation allows provenance to be stored in a standard, machine accessible form that can be shared and queried. These tools can also relate context information available in the system to provide semantic value to provenance.

3.5 Provenance Dissemination

In order to use provenance, a system should allow rich and diverse means to access it. A common way of disseminating provenance data is through a derivation graph that users can browse and inspect. Additional metadata about the datasets in the provenance can also be provided through such a tool to enhance its usefulness. Users can also search for datasets based on their provenance metadata, such as to locate all datasets generated by a executing an erroneous workflow or to find the owners of all source data used to derive a certain data. If semantic provenance information is available, results of queries over provenance can automatically feed input datasets for a workflow at runtime. The derivation history of datasets can be used to replicate data at another site, or update it if a dataset is stale due to changes made to its ancestors. Provenance retrieval APIs can additionally allow users to implement their own mechanism of usage.

4 Survey of Data Provenance Techniques

In our survey of data provenance, we identified nine major works that, taken together, provide a comprehensive overview of research in this field. While additional works on provenance were reviewed, they were discarded because the treatment of provenance is incidental, not the prime research focus, or presented use-cases and concepts without a design or prototype. While Lanter et al and Woodruff et al provide a historical basis of lineage in scientific systems, the rest of the projects represent current research on data provenance. A summary of their characteristics can be found in **Table 1**.

4.1 Lanter, D. P. (Lineage Information Program)

Lanter made pioneering contributions in the early 1990's to the study of lineage in GIS applications. Lineage gives a notion of the quality of GIS datasets based on the source data that went into deriving it [5]. This quality indication helps GIS users to determine the fitness of use of the data for their application [55]. GIS applications use a cartographic model to transform and derive *spatial layers*. These spatial layers form the basic dataset in GIS, and have metadata attributes that describe the transformation algorithm, the properties of the layers, and the intended use of the data. The attributes follow cartographic standards for spatial data set, such as the Spatial Data Transfer Standard (SDTS), that have associated quality and lineage statements.

Lanter designed a Lineage Information Program (LIP) [5, 55] for spatial databases that models lineage as a semantic network. A *semantic network* is a bi-directional knowledge representation graph where nodes and edges are given domain-specific semantic meaning [55]. For the GIS domain, nodes represent map layers and edges denote a parent-child relationship between layers such that a child layer is derived from the parent layer through some processing steps. Layers that do not have parents are *source layers* and they originate from outside the database. *Product layers* do not have children and are the end result of a GIS application. *Intermediate layers* are generated in the process of transforming source layers to a product layers. In addition to parent-child links, LIP uses a data structure called *frame* which describes the metadata of a spatial layer. Three types of frames are available: *source frame*, containing quality information about the source layers, such as scale and projection; *command frame*, with the commands used to derive intermediate and product layers; and *product frames* that has metadata specific to the product layers, like the release date and manager.

Users transform and create layers by the executing commands in a spatial database shell [55]. LIP intercepts these data manipulation commands before they reach the spatial database and parses them to identify the input and output layers affected by this command. It then records the relationship between the layers by creating a parent-child link between them and associates a command frame, containing the executed shell command, with the output layer [5]. LIP follows a data-oriented provenance technique since the parent-child link directly relates the two data layers, and the command frame containing the derivation step is tied to the derived spatial layer. Source and product frames are populated by users when they import source layers or export product layers from the spatial database respectively. The granularity for LIP is at the level of spatial layers. The frames and the parent-child links of the semantic network are stored in a meta-database separate from the spatial database. LIP is intended for use with individual spatial databases and does not address scalability as an issue.

Information in the lineage meta-database can be interrogated interactively using command-line queries, and LIP traverses the stored semantic network to answer the queries. Users can, for example, query for the complete lineage of a layer in the database or locate all layers derived from a certain source layer. The lineage information for a layer can be used to regenerate it automatically if its ancestral layers were modified [85]. As an extension to this, the storage space for a spatial database can be optimized by removing all derived data layers and retaining only the source layers, with the derived layers being generated just-in-time when requested for [86]. Lineage can also identify equivalent layers in order to remove redundancy in the database [85].

4.2 Chimera & the Virtual Data Grid (VDG)

Chimera [45] is a prototype implementation of a Virtual Data Grid (VDG) that manages the derivation and analysis of data objects in collaboratory environments [64]. Chimera provides a generic solution for scientific communities, such as high-energy physics (GriPhyN) and astronomy (SDSS), which have data-intensive needs. Chimera tracks provenance in the form of the data derivation steps for datasets and uses it for on-demand regeneration of derived data ("virtual data"), comparison of datasets, and audit of data derivations [45].

Chimera uses a process-oriented model to record provenance. Users construct workflows (called *derivation graphs* or DAGs) using a high-level Virtual Data Language (VDL) [45, 64]. The VDL conforms to a schema that represents data products as abstract typed *datasets* and their materialized *replicas* that are available at a physical location. Datasets can be files, tables, and objects of varying granularity (though the prototype supports only files), and the *type* determines their structure and representation. Computational process templates, called *transformations*, are scripts in the file system (in future, web services) that transform typed datasets [64]. The parameterized instance of the transformations, called *derivations*, can be connected to form workflows that consume and produce replicas. These derivation workflows are scheduled and executed on the Grid using Globus Toolkit v2, whereupon they create *invocation* objects for each derivation in the workflow. The invocation records are automatically annotated with runtime information of the process. Invocation objects are the glue that link input and output data products, and they constitute an annotation scheme for representing the provenance. Semantic information on the dataset derivation is not collected.

The lineage in Chimera is represented in VDL that is managed by a virtual data catalog (VDC) service. The VDC maps the VDL to a relational schema and stores it in a relational database accessible through SQL queries [45]. Metadata can be stored in a single VDC, or distributed over multiple VDC repositories that enables scaling through federation. VDL can contain inter-catalog references to refer to data and processes in external VDCs. Lineage information can be retrieved from the VDC using queries written in VDL that can, for example, recursively search for derivations that generated a particular dataset. A virtual data browser that uses the VDL queries to interactively access the catalog is proposed [45]. A novel use of provenance in Chimera is to plan and estimate the cost of regenerating datasets. When a dataset has been previously created and it needs to be regenerated (e.g. to create a new replica), its provenance guides the workflow planner in selecting an optimal plan for resource allocation [45, 64].

4.3 myGrid

The myGrid project provides middleware in support of *in silico* experiments in biology, modeled as workflows in a Grid environment [87]. *In silico* experiments are web-based data analysis and computational procedures analogous to laboratory experiments [46, 73]. Bioinformatics places a higher importance on semantically enhanced information model as opposed to availability of computationally intensive resources [87]. myGrid services include resource discovery, workflow enactment, and metadata and provenance management, that enable information integration and help address the semantic complexity.

myGrid is service-oriented and executes workflows written in *XScufl* language using the *Taverna* workflow environment [74]. A log of the workflow enactment contains the services invoked, their parameters, the start and end times, the data products used and derived, and ontology descriptions, and it is automatically recorded when the workflow executes. This process-oriented workflow derivation log is inverted to infer the provenance for the intermediate and final data products [46]. Users need to annotate workflows and services with semantic descriptions to enable this inference and have the semantic metadata carried over to the data products, and this is part of the user overhead.

In addition to contextual and organizational metadata such as owner, project, and experiment hypothesis, users can provide domain specific ontological terms to describe the data and the experiment [6, 46]. These contextual information help scientists to understand the experiments that have been conducted and to validate them. Scientists can also directly upload datasets into the myGrid environment along with their associated provenance information [6]. XML, HTML, and RDF are used to represent syntactic and semantic provenance metadata using the annotation scheme [61]. The granularity at which provenance can be stored is flexible and is any resource identifiable by an LSID [74]. This includes provenance about components such as experiments, workflow specifications, and services, in addition to data products [61].

The myGrid Information Repository (mIR) data service is a central repository built over a relational database to store metadata about experimental components [74]. A number of ways are available for

knowledge discovery by reasoning over semantically enhanced provenance logs [73]. A rudimentary viewer is available in the form of Launchpad, a generic visualizer for LSID environments [46]. The semantic provenance information available as RDF can be viewed as a labeled graph using the Haystack semantic web browser [74]. COHSE (Conceptual Open Hypermedia Services Environment), a semantic hyperlink utility, is another tool used to build a semantic web of provenance. Here, semantically annotated provenance logs are interlinked using an ontology reasoning service and displayed as a hyperlinked web page. Provenance information generated during the execution of a workflow can also trigger the rerun of another workflow whose input data parameters it may have updated [87].

4.4 Collaboratory for Multi-scale Chemical Science (CMCS)

The CMCS project is an informatics toolkit for collaboration and metadata-based data management for multi-scale science [79, 88]. CMCS manages heterogeneous data flows and metadata across multi-disciplinary sciences such as combustion research where it is not possible to enforce metadata standards across the domain [47]. Provenance metadata in CMCS supplements the traditional means of establishing the pedigree of data through scientific publications [89].

CMCS uses the Scientific Annotation Middleware (SAM) repository for storing resources represented as URL referenceable files and collections [79]. CMCS uses an annotation scheme to associate XML metadata properties with the files in SAM, and manages them through a Distributed Authoring and Versioning (WebDAV) interface. Files form the level of granularity and all resources such as data objects, processes, web services, and bibliographic records are abstracted as files. The type of the resource is distinguished by the Multi-purpose Internet Mail Extension (MIME) type for that file [88]. Dublin Core (DC) [90] elements like *Title*, *Creator*, and *Date* are used as XML properties to describe general characteristics of the resources. Additionally, DC verbs like *Has Reference*, *Issued*, and *Is Version Of* semantically relate resources with each other through XLink references in SAM [88]. These verbs capture the provenance by relating data files with their deriving processes. Direct association of provenance metadata with the data object makes this a data-oriented model. Heterogeneous metadata schemas are supported by mapping them to standard DC metadata terms using XSLT translators associated with specific MIME types. One limitation of exclusively using DC terms as a semantic metadata scheme is that they are intended for generic use with any type of resource. Hence the semantic meaning conveyed by the terms depends on the context they are used in and the resource they describe, leaving scope for ambiguity.

CMCS does not provide a facility for automated collection of lineage from a workflow's execution. Data files and their metadata are populated by DAV-aware applications in workflows or manually entered by scientists through a portal interface [79]. Provenance metadata properties can be queried from SAM using generic WebDAV clients. Special pedigree browser portlets allow users to traverse the provenance metadata for a resource as a web page with hyperlinks to related data, or by visualizing a labeled graph represented in the Graphics eXchange Language (GXL). Provenance information can also be exported to RDF that semantic agents can use to infer relationships between resources. Provenance metadata that indicate data modification can generate notifications that trigger workflow execution to update dependent data products.

4.5 Provenance Aware Service-oriented Architecture (PASOA)

The Provenance Aware Service Oriented Architecture (PASOA) project is building a provenance infrastructure for recording, storing and reasoning over provenance using an open provenance protocol that will foster interoperability among e-science communities [8, 84]. PASOA identifies several requirements for a provenance system in a service oriented architecture, such as *verifiability* of actors involved in a process and *reproducibility* of the process, *accountability* and *preservation* of provenance over time, *scalability* of the provenance system, *generality* to support diverse Grid applications as well as *customizability* as required [8, 84]. *Actors* are either clients of services or services that are invoked, and they generate two kinds of provenance during a workflow's execution. *Interaction* provenance, describing

the input and output parameters of a service invocation, is generated and corroborated by both actors – client and service – in the invocation. *Actor* provenance is metadata about the actor's own state during a service invocation (e.g. CPU usage or the service script) and is not verifiable [44, 84].

The Provenance Recording Protocol (PReP) defines fourteen interaction provenance messages that are generated by the actors, synchronously or asynchronously, with each service invocation. They are divided into four phases: *negotiation phase*, *invocation phase, provenance recording phase*, and *termination phase*, during which the actors agree upon a provenance service to record the provenance, perform the service invocation, record their interaction provenance, and terminate the protocol, respectively. All interaction and actor provenance messages generated by the clients and services in a workflow are correlated using an *ActivityID* present in the provenance messages. These form a process oriented provenance trace of the workflow recorded as annotations, and data provenance needs to be independently derived by linking all assertions that have the same ActivityID as that of the assertion containing the data as output. The granularity of the provenance is at the level of the input and output parameters to the web service.

Provenance Recording for Services (PReServ) is a web service implementation of the PReP protocol that stores the provenance either in memory, in a relational database, or in the file system [44]. The actual representation of provenance is not apparent. A performance overhead of 10% has been observed when the provenance assertions are submitted asynchronously by the actors, with the overhead increasing when the provenance is sent synchronously with the service invocation. Overhead also lies in modifying the actors to generate the provenance messages, though this can be reduced by just modifying the workflow engine to generate provenance [91]. Methods to scale the provenance store through federation are being considered.

A querying interface is not defined as part of the PReP protocol but a basic querying API is available to retrieve provenance from PReServ [44]. Basic queries to locate all data that were derived using the same service can be performed. Semantic validity checking of services and their inputs/outputs is possible by comparing the expected inputs/outputs of a service, available in a semantic registry, with the actual inputs/outputs available with the provenance. Other uses, such as repeating a workflow using the inputs to services available as provenance, are also foreseen.

4.6 Earth System Science Workbench (ESSW)

The Earth System Science Workbench (ESSW) [92] is a metadata management and data storage system for earth science researchers. ESSW is used to manage custom satellite-derived data products and compose relevant metadata for an ecological research project that spans multiple organizations [80]. Lineage is a key facet of the metadata created in the workbench, and is used for detecting errors in derived data products and in determining the quality of datasets.

ESSW uses a scripting model for data processing i.e. all data manipulation is done through scripts that wrap existing scientific applications [80]. The sequence of invocation of these data transformation scripts by a master workflow script forms a DAG. Data products at the granularity of files are consumed and produced by the scripts, with each data product and script having a uniquely labeled metadata object. As the workflow script invokes individual scripts, these scripts, as part of their execution, compose XML metadata for themselves and the data products they generate using script libraries that are provided. The workflow script links the data flow between successive scripts using their metadata ids to form the lineage trace for all data products, represented as annotations. By chaining the scripts and the data using parent-child links, ESSW is balanced between data and process oriented lineage.

ESSW puts the onus on the script writer to record the metadata and lineage using templates and libraries that are provided. The libraries store metadata objects as files in a web accessible location and the lineage separately in a lineage server that uses a relation database backend [80]. This separation of the metadata and the lineage repository enables legacy metadata systems to record lineage information without modifying their existing metadata management methods. Scalability is not currently addressed though

there is a proposal to federate lineage across organizations [80]. *Linchpin* data products that exist at the boundaries of two workflows (i.e. the data output of one workflow is the input to the other) running at different organizations are identified and link the lineage chains in both the organizations.

The metadata and lineage information can be navigated as a workflow DAG through a web browser that uses PHP scripts to access the lineage database [92]. Future work includes encoding lineage information semantically as RDF triples to help answer richer queries [58, 80]. The data products and deriving scripts form the subject and object in the RDF triple and verbs such as *createdBy* and *inputTo* establish the relation between them.

4.7 Tioga

Tioga [93] is a database centric visualization tool, built on top of POSTGRES, that uses a "drag and drop" approach to construct database programs. User-defined POSTGRES functions with a typed inputs and outputs are visually depicted as boxes that are interconnected through arrows representing the dataflow between them, effectively forming a workflow. Tioga tracks fine-grained data lineage within the database and is motivated by the needs of atmospheric scientists to trace the steps that led to identification of cyclone signatures based on attributes in the database [53].

Tioga was one of the earliest systems to represent provenance using inverse functions registered for user-defined functions (UDFs) [53]. For a UDF $\boldsymbol{f}$ that takes as input a set of tuples $\boldsymbol{I^1}$ from a certain table and generates a set of output tuples $\boldsymbol{I}$, the inverse function is given by the function $\boldsymbol{f^1}$ that takes as input the tuples $\boldsymbol{I}$ and generates the source tuples $\boldsymbol{I^1}$. UDFs that do not have exact inverse functions can instead register *weak-inverse* functions that provide a subset or superset of the source tuples. A weak inverse function $\boldsymbol{f^w}$ takes the tuples $\boldsymbol{I}$ as input and generates the tuples $\boldsymbol{I^w}$, that are either *complete* ($\boldsymbol{I^w} \supseteq \boldsymbol{I^1}$) or *pure* ($\boldsymbol{I^w} \subseteq \boldsymbol{I^1}$) or *both* ($\boldsymbol{I^w} = \boldsymbol{I^1}$). The accuracy of $\boldsymbol{I^w}$ can be enhanced by defining *verification* functions, $\boldsymbol{f^v}$, that take $\boldsymbol{I}$ and $\boldsymbol{I^w}$ as inputs to give the tuples $\boldsymbol{I^v}$, that are closer to the exact inverse set $\boldsymbol{I^1}$. They too have the properties of complete ($\boldsymbol{I^v} \supseteq \boldsymbol{I^1}$) or pure ($\boldsymbol{I^v} \subseteq \boldsymbol{I^1}$) or both ($\boldsymbol{I^v} = \boldsymbol{I^1}$).

The weak-inverse and verification functions registered for UDFs, when executed, track the output tuples of the UDFs back to their source tuples, and provide an approximate version (unless the functions are complete as well as pure) of the lineage [53]. Weak-inverse and verification functions for UDFs can further be decomposed and represented as the union of weak-inverse and verification functions of each attribute in the tuples generated by the UDF, thus achieving attribute-level granularity. The weak-inverse and verification functions for a single attribute type can be reused by all tuples having that attribute type thus reduceing the overhead for defining these functions. It is possible that the UDFs may not even have non-trivial weak-inverse and verification functions. Associating the functions directly with the data makes this a data oriented model. Despite fine-grained data provenance, the storage overhead is low since defining a weak-inverse and a verification function for each attribute type denotes the lineage for all tuples that contain those attributes types and hence the systems scales with the number of tuples.

The lineage information provided by Tioga is limited since the inputs that went into creating that data item are alone recorded in the absence of any other semantic information. Fine-grained lineage can be supplemented by coarse-grained metadata techniques if additional provenance information is required [53]. The weak-inverse and verification functions may be modeled as boxes in Tioga whose results, that form the lineage, can be visualized [93]. Users can also define database queries that use these inverse functions to extract the lineage. Since inverse-function strategies are executed just-in-time, there may be significant computational overhead to retrieve the lineage.

4.8 Buneman, P.

Buneman has presented a collection of theoretical work on managing provenance and on the related topics of annotations, archiving, and versioning in scientific databases. Scientific databases are often "curated" by adding annotations to them [4, 94], and in sciences like biology and astronomy, there are curated source databases which are processed by scientists to create new datasets [95]. Provenance is

necessary to track the processed datasets back to the curated source database, as well as to propagate additional annotations on the derived data back to the source database [38]. For databases that change over time, compact versioning is essential to recover data referenced by the lineage of data derived from an earlier version of the database [94].

Buneman puts forth two forms of data provenance that he terms "why" provenance and "where" provenance [4]. *Why* provenance gives the reason the data was generated, say, in the form of a proof tree that locates source data items contributing to its creation. *Where* provenance provides an enumeration of the source data items that were actually copied over or transformed to create this data item. These provenances are defined for relational, object oriented, and semi-structured databases that satisfy a certain deterministic data and query model. The *why* provenance for a data created by a query on the database is the collection of values, called the *minimal witness basis*, that prove the query output and can be found by a pattern matching algorithm. A similar algorithm to generate the *where* provenance, called the *derivation basis*, is also available. These algorithms are akin to determining lineage through query inversion. The algorithm to determine *why* provenance is invariant under rewriting of the original query, while only the class of *traceable queries* are rewriteable without affecting the *where* provenance algorithm [4]. The inverse queries are associated with the derived data making it data oriented provenance with granularity at the level of tuples and attributes in the database.

Why and *where* provenance are respectively analogous to two view update problems in relational databases [38]. The *view deletion problem* requires the identification of the smallest set of tuples in the database whose removal will cause a given tuple in a view to be deleted while minimizing deletion of other tuples in the view. The *annotation placement problem* needs to locate the attributes in the source database that need to be annotated so that the annotation will appear in an attribute of the view, while minimizing side effects to other attributes in the view. An investigation of the computational complexity of solving these problems for views created by SPJU (select-project-join-union) relational operators concludes that for views created using PJ and JU queries, finding their *view deletions* without side effects on the source or view data is NP-hard, while for SPU and SJ views, *view deletions* without side effects are polynomial time solvable. Similarly, for solving the *annotation placement problem*, PJ views are intractable while SJU and SPU are tractable.

4.9 Cui, Y. & Widom, J. (Trio)

Cui and Widom [33, 51] trace lineage information for view data and general transformations in data warehouses. The Trio project [54] leverages some of this work in a proposed database system which has data accuracy and data lineage as inherent components. While data warehouse mining and updation motivates lineage tracking in this project, any system that uses database queries and functions to model workflows and data transformations can apply such techniques.

Identifying the maximal set of tuples from source tables, that produced a data item in a materialized warehouse view is defined as the *view data lineage* problem [51, 96]. A database view can be modeled as a query tree that is evaluated bottom-up, starting with leaf operators having tables as inputs and successive parent operators taking as input the relations resulting from its child operators [51]. For ASPJ (Aggregate-Select-Project-Join operator) views, it is possible to automatically create an inverse query for the view query that will identify the tuples from the source tables that form the view data's lineage [51]. A *Split* algorithm that operates on the materialized view results and the source tables is recursively applied to each segment of the canonicalized ASJP query tree in a top-down manner [51]. A union of the results of all the *Split* operations returns the exact tuples in the source tables that is the required lineage for the view. Computing the lineage trace can be optimized by using auxiliary tables that cache source tables and intermediate results of the ASJP view query evaluation [51]. This reduces costly access to source tables external to the warehouse and prevents inconsistencies due to updates in the source tables. Comparison for ten auxiliary table schemes, ranging from storing no auxiliary table to storing all source tables, is provided [33].

Trio [54] uses this inversion model to automatically determine the source data for tuples created by view queries. The inverse queries are recorded at the granularity of a tuple and stored in a special *Lineage* table that records, for each tuple, its creation *timestamp*, the *derivation-type* (such as by a view query, an insert or update query, or user defined functions), and additional lineage related data. This direct association of lineage with tuples makes this a data-oriented provenance scheme. While lineage for view tuples can be derived using the above algorithm, mechanisms to handle non-view tuples are yet to be determined. Lineage in Trio is simply the source tuples and the view query that created the view tuple, with no semantic metadata recorded. Scalability is not specifically addressed either. Other than querying the

Lineage table, some special purpose constructs for retrieving lineage information through a Trio Query Language (TriQL) are planned. Since recording attribute level accuracy is also an inherent part of Trio, queries combining both lineage and accuracy information shall also be supported.

5 Discussion

The survey we presented exposes interesting open research questions on provenance and challenges that need to be overcome to make provenance pervasive in the broader community. Data is increasingly being shared across organizations and it is essential for provenance to be shared along with the data. Most of the projects surveyed have their own proprietary protocols for managing provenance, and the absence of open standards for collecting, representing, storing, and querying for provenance is an obvious hindrance to promoting interoperability. Open standards will also promote federated collection of provenance from actors across organizations instead of a centralized approach where, say, a workflow engine is solely responsible for recording provenance. Standards will also allow provenance collection to be pushed into the middleware instead of expecting the user or service provider to do it. The work by PASOA on defining a provenance recording API [8] is in the right direction but needs further refinement on how provenance is represented and queried. Any such standard will have to satisfy the diverse needs of the multifarious scientific or business domains in which it will be used, and these requirements need to be identified.

Standardizing semantic terms to describe provenance will allow unambiguous interpretation of provenance [46]. This, coupled with domain specific ontologies, will allow automated verification of the provenance and enable richer queries. Such a verification can eventually be extended to any piece of information present in the semantic web [97] through its provenance [62]. myGrid is progressing along these lines by migrating to the Web Ontology Language (OWL) for describing their provenance. CMCS too has preliminary support for a semantic description of provenance that can be improved upon by using specific semantic terms to describe provenance instead of using overloaded Dublin Core verbs.

Using provenance as a basis for decision making largely depends upon the trustworthiness of provenance. Assertions about the pedigree of data will have to be backed by the identity of a trusted person or organization to make the assertion meaningful [98]. There can also be multiple versions of truth provided by different entities involved in the data derivation and these will have to be mediated [44, 59]. Also, there should be assurances that the provenance information was not tampered with and signing provenance using digital signatures is a solution. Providing such convincing assertions on provenance will enhance its value and lead to a broader use of provenance for decision making, that in turn will promote a wider collection of provenance.

The granularity of provenance depends on the discipline in which it is collected and the application in which it is used. Developing standards to represent provenance is related to the issue of naming datasets uniquely and uniformly so that they can be referenced by the provenance [99]. Naming schemes like LSID that use URNs or URIs to identify datasets allow provenance to refer to abstract datasets irrespective of their granularity or representation. At the same time, care should be taken that these indirection schemes do not prevent us from discovering or indicating relationships between the provenances of datasets, as may be the case between the provenance of a data collection and the provenance of a data entry that is present in the collection.

There is potential for convergence of lineage tracking in databases and provenance collection in service-oriented architectures. Databases are moving towards supporting service invocations from within queries and web services use databases to store and transform data. When data passes through multiple organizations, it is possible that is processed by both such systems, motivating the need to tracking provenance seamlessly across these architectures transparently. Federated databases may pose additional challenges due to the source transparency they provide over heterogeneous data [52].

Annotations and inversion are two design choices to represent provenance, though this choice may be limited by the way data is processed and due to restrictions on the kinds of transformations that can be inverted. Inversion seems to be more optimal from a storage perspective and may be preferred by organizations requiring provenance tracking for a large number of fine-grained datasets. But all the systems we surveyed that use the inversion technique [4, 51, 53] require the source data to be available in order to execute the inverse queries. If the source data keeps changing, this may require a significant amount of auxiliary data to be retained to determine the lineage and this offsets the storage benefits of inversion. A deeper study of the storage needs of the inversion techniques will help architects of provenance systems to make a design decision between annotation and inversion, and fuel further research into the inversion technique.

Another design choice is whether to use a process oriented or data oriented model of provenance. This has underpinnings on the cost of executing relevant queries on the provenance metadata. It is potentially costlier to extract data provenance from a process oriented model since this involves examining all process oriented provenance records in which this data appears and selecting those that led to the data's creation. Data oriented provenance would provide this information immediately. On the other hand, data oriented provenance may have a costly overhead to execute a process related query, say one that locates workflows in which a particular data was used. Data oriented provenance allows the data provenance to be represented and ported in a self-contained manner.

Efficiently federating the collection, storage, and retrieval of provenance is necessary for it to scale across communities. Negligible research has been done on scaling provenance systems but one can expect existing distributed architectures such as federated registries and peer-to-peer systems to guide this development.

Retaining lineage about data even after it was deleted, or lineage that traces the reason for a data being deleted (as opposed to how it was created) is termed phantom lineage [54, 75]. This tracks the lifetime of a data from its creation to deletion and finds interesting uses such as for auditing. Provenance can be used to estimate quality metrics for data or for deriving new hypothesis from the provenance of experimental results. Discovering such novel ways to use provenance will drive more organizations to collect provenance. For this to happen, provenance needs to be fully understood and studied in the context of its potential use in each domain. Many of its current applications are largely generic in nature and there remains potential to use provenance in much better ways.

6 Conclusion

In this paper, we presented a taxonomy to understand and compare provenance techniques used in e-science projects. The exercise shows that provenance is still an exploratory field and several open research questions are exposed. There need to be means to guarantee the source of provenance and assert its truthfulness in order for provenance to be useful beyond an individual organization [59]. Ways to federate provenance information is essential for scalable collection, storage, and retrieval of provenance. Evolution of common metadata standards, semantic terms, and service interfaces to manage provenance in diverse domains will also contribute to a wider adoption of provenance and promote its sharing [63]. The ability to seamlessly represent provenance of data derived from both workflows and databases can help in its portability. Ways to store provenance about missing or deleted data (phantom lineage [54]) require further consideration. Finally, a deeper understanding of provenance is needed to identify novel ways to leverage it to its full potential.

7 Acknowledgements

The authors would like to acknowledge the anonymous reviewers of the SIGMOD Record September 2005 issue (in which a shorter version of this paper appears) for raising several interesting points and that we've tried to address in this paper. We'd also like to thank Prof. Edward Robertson of Indiana University for his valuable comments on a presentation of this survey.

Figures and Tables

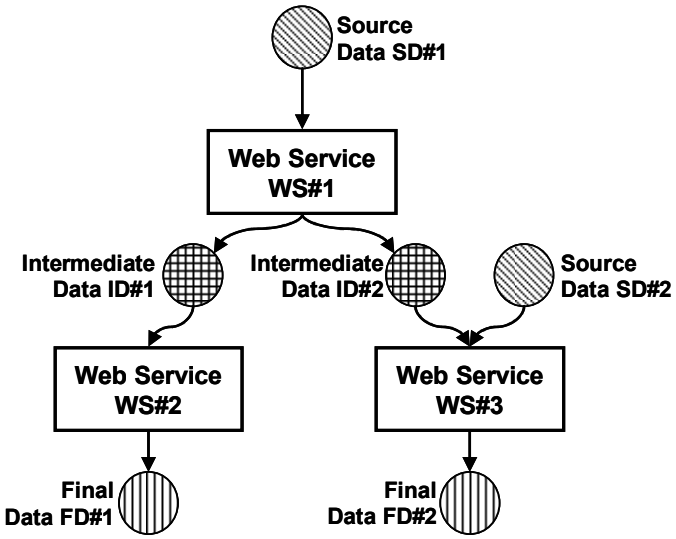


Figure 1 A Workflow execution represented as a directed acyclic graph (DAG). There are three web services labeled WS#1, WS#2, and WS#3 that consume and generate data products. The provenance for the source data SD#1 and SD#2 is not defined; for an intermediate data, say ID#1, the provenance includes SD#1 and WS#1; and for a final data, say, FD#2, the lineage is SD#1, WS#1, ID#2, SD#2, WS#3.

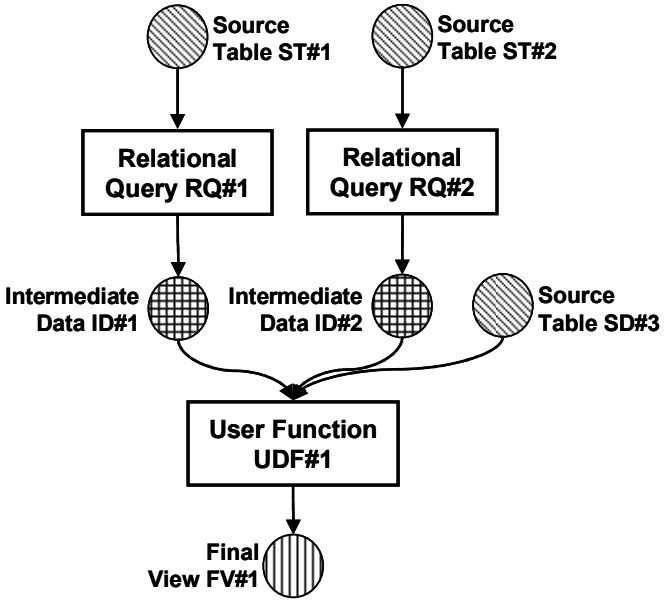
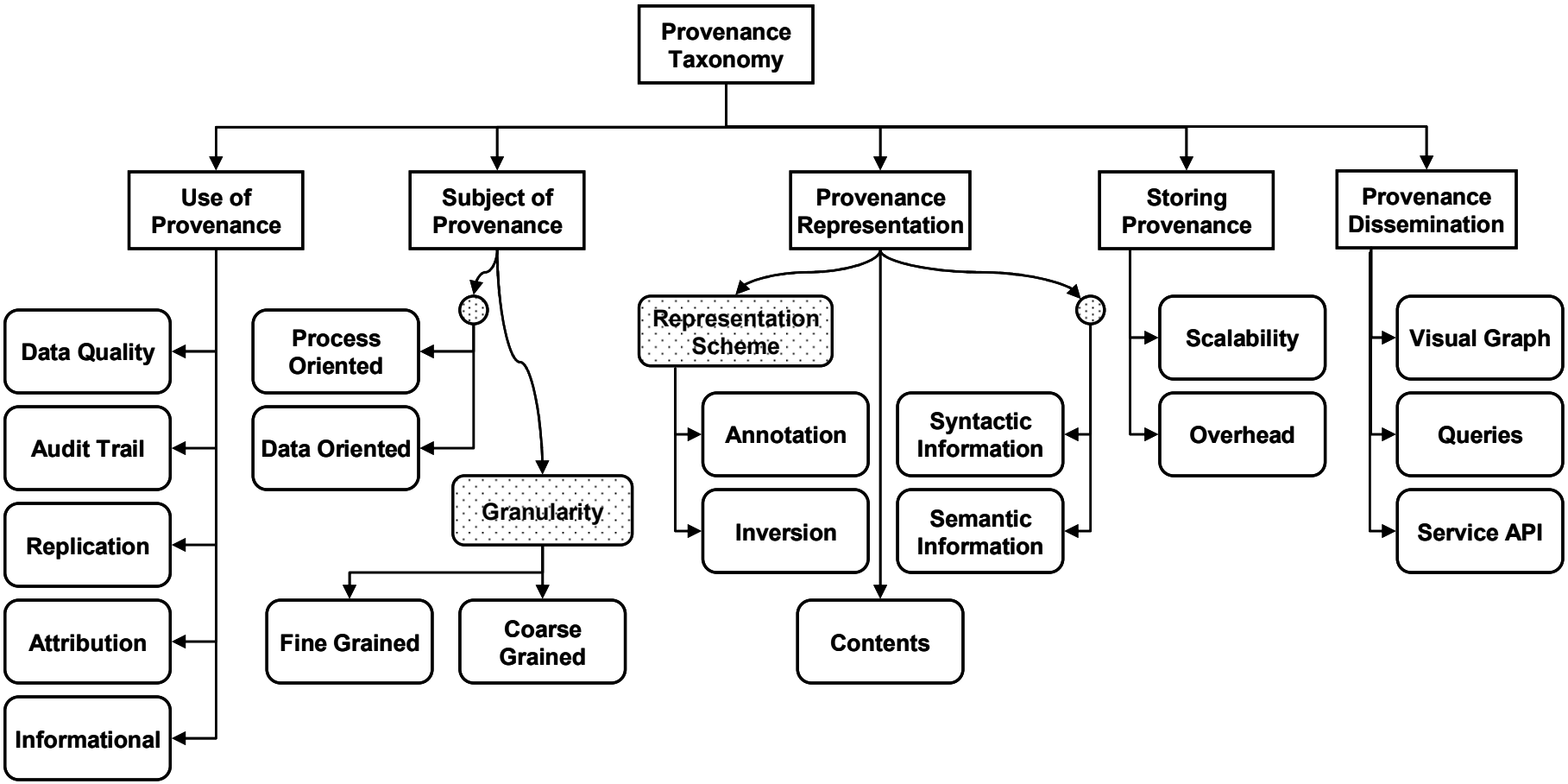


Figure 2 Database query tree analogous to the Workflow DAG in **Figure 1**. Here, the final view, FV#1, is derived from external sources and stored in a data warehouse. The lineage for this view, FV#1, depends on the source tables ST#1, ST#2, and ST#3, on the intermediate views, IV#1 and IV#2, and the relational queries, RQ#1, and RQ#2, and the user defined function, UDF#1. The exact tuples from the tables and views that contributed to the final view depends upon the queries and the function.

Figure 3 The Provenance Taxonomy



	Lanter, D. P. (LIP)	Chimera	MyGRID	CMCS	PASOA	ESSW	Tioga	Buneman, P.	Cui, Y., Widom, J. (Trio)
Applied Domain	GIS	Physics, Astronomy	Biology	Chemical Sciences	Biology	Earth Sciences	Atmospheric Science	Generic (Scientific databases)	Generic
Data Processing Framework	Command Processing	Service Oriented	Service Oriented	Service Oriented	Service Oriented	Script Based	Relational Database	Relational/Semi Structured Database	Relational Database
Application of Provenance	Informational; update stale, regenerate & compare data	Informational; Audit; Data Regeneration; Planning	Contextual Information; Re-enactment	Informational; Data Update	Informational; Re-enactment	Informational	Informational; Track errors	Annotation propagation; View Updation	Information; update propagation
Data/Process Oriented	Data	Process	Process	Data	Process	Both	Data	Data	Data
Granularity	Spatial layers	Abstract datasets (Currently files)	Abstract resources having LSID	Files	Abstract parameters to Workflow	Files	Attributes in Database	Attributes & Tuples in Databases	Tuples in Database
Representation Scheme	Commands & Frames as Annotations	Virtual Data Language Annotations	XML/RDF Annotations	Dublin Core XML Annotations	Annotations	XML/RDF Annotations	Inverse Functions	Inverse Queries	Inverse queries
Semantic Information	No	No	Yes	Limited	No	No, Proposed	No	No	No
Storage Repository/ Backend	MetaDatabase	Virtual Data Catalog/ Relational DB	mIR repository/ Relational DB	SAM over WebDAV/ Relational DB	PReServ/ Relational DB, File System	Lineage Server/ Relational DB	Relational DB	N/A	Relational DB
Provenance Collection Overhead	Store User commands; solicit metadata	User defines derivations; automated WF trace	User defines service semantics; Automated WF Trace	Manual; Apps use DAV APIs, Users use portal	Manual; Actors use PReP API	Libraries assist user to generate,store provenance	User registers inverse functions	N/A	Inverse queries automatically generated
Addressed Scalability	No	Yes	No	No	No (Proposed)	No (Proposed)	Yes	N/A	No
Provenance Dissemination	Queries	Queries	Semantic browser; Lineage graph	Browser; Queries; GXL/RDF	Queries	Browser	Queries; box-and-arrows visualization	N/A	SQL/TriQL Queries

Table 1 Summary of characteristics of surveyed data provenance techniques

References

[1] "Large Hadron Collider Computing Grid," *http://lcg.web.cern.ch/LCG/index.html*, visited 2005.

[2] "Sloan Digital Sky Survey," *http://www.sdss.org*, 2005.

[3] P. S. Wadhwa and P. Kamalapur, "Customized Metadata Solution for a Data Warehouse - A Success Story," *Wipro Technologies White Paper*, 2003.

[4] P. Buneman, S. Khanna, and W. C. Tan, "Why and Where: A Characterization of Data Provenance," in *ICDT*, 2001, pp. 316-330.

[5] D. P. Lanter, "Design Of A Lineage-Based Meta-Data Base For GIS," in *Cartography and Geographic Information Systems*, vol. 18, 1991, pp. 255-261.

[6] M. Greenwood, C. Goble, R. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn, "Provenance of e-Science Experiments - experience from Bioinformatics," in *Proceedings of the UK OST e-Science second All Hands Meeting*, 2003.

[7] R. Bose and J. Frew, "Lineage retrieval for scientific data processing: a survey," in *ACM Comput. Surv.*, vol. 37. New York, NY, USA, 2005, pp. 1--28.

[8] S. Miles, P. Groth, M. Branco, and L. Moreau, "The requirements of recording and using provenance in e-Science experiments," in *Technical Report*: Electronics and Computer Science, University of Southampton, 2005.

[9] "Workshop on Data Derivation and Provenance, Chicago," P. Buneman and I. Foster, Eds., 2002.

[10] "Workshop on Data Provenance and Annotation, Edinburgh," D. Berry, P. Buneman, M. Wilde, and Y. Ioannidis, Eds., 2003.

[11] K. Renaud, "Data Provenance and Annotation Resource Home Page http://www.dcs.gla.ac.uk/~karen/Provenance," Department of Computer Science, University of Glasgow, 2005.

[12] "eBank UK study of provenance http://www.ukoln.ac.uk/projects/ebank-uk/provenance," eBank UK, 2005.

[13] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," in *International Journal of Supercomputer Applications.*, vol. 15, 2001.

[14] J. Brase, "Using Digital Library Techniques - Registration of Scientific Primary Data," in *Lecture Notes in Computer Science*, vol. 3232, 2004, pp. 488-494.

[15] "The Proposed Standard for Digital Cartographic Data," in *The American Cartographer*, vol. 15: Task Force on Digital Cartographic Data Standards, 1988.

[16] "Spatial Data Transfer Standard (SDTS)," in *ANSI NCITS 320-1998*: National Committee for Information Technology Standards (NCITS), 1998.

[17] "Geodetic Control," *Wisconsin State Cartographer's Office and University of Wisconsin, Madison*, 2004.

[18] D. G. Clarke and D. M. Clark, "Lineage," in *Elements of Spatial Data Quality*, S. C. Guptill and J. L. Morrison, Eds., 1995.

[19] J. L. Romeu, "Data Quality and Pedigree," in *Material Ease*, 1999.

[20] H. V. Jagadish and F. Olken, "Database Management for Life Sciences Research," in *SIGMOD Record*, vol. 33, 2004, pp. 15-20.

[21] D. B. Searls, "Data integration: challenges for drug discovery," in *Nature Reviews Drug Discovery*, vol. 4, 2005, pp. 45-58.

[22] "Access to genetic resources and Benefit-Sharing (ABS) Program," Institute for Advanced Studies, United Nations University, 2003.

[23] S. G. Djorgovski, R. R. Gal, S. C. Odewahn, R. R. d. Carvalho, R. Brunner, G. Longo, and R. Scaramella, "The Palomar Digital Sky Survey (DPOSS)," *Wide Field Surveys in Cosmology*, 1998.

[24] J. Gray and A. Szalay, "The World-Wide Telescope, an Archetype for Online Science," *Microsoft Research Technical Report MSR-TR-2002-75*, 2002.

[25] B. Mann, "Some Data Derivation and Provenance Issues in Astronomy," in *Workshop on Data Derivation and Provenance, Chicago*, 2002.

[26] N. I. Hachem, Ke~Qiu, M. A. Gennert, and M. O. Ward, "Managing Derived Data in the Gaea Scientific DBMS," in *VLDB*, 1993, pp. 1-12.

[27] J. v. d. Hoven, "Data Architecture: Standards for the Effective Enterprise," *Information Systems Management*, 2004.

[28] M. Hanrahan, "The Essential Ingredient: How Business Intelligence depends on data quality," *Dal Cais Research White Paper*, 2004.

[29] W. H. Inmon, "The Data Warehouse and Data Mining," in *Communication of the ACM*, vol. 39, 1996, pp. 49-50.
[30] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," in *SIGMOD Record*, vol. 26, 1997, pp. 65-74.
[31] P. A. Bernstein and T. Bergstraesser, "Meta-Data Support for Data Transformations Using Microsoft Repository," in *IEEE Data Engineering Bulletin*, vol. 22, 1999, pp. 9-14.
[32] D. Agrawal, A. E. Abbadi, A. Mostfaoui, M. Raynal, and M. Roy, "The Lord of the Rings: Efficient Maintenance of Views at Data Warehouses," in *DISC*, 2002, pp. 33-47.
[33] Y. Cui and J. Widom, "Lineage tracing for general data warehouse transformations," in *VLDB Journal*, vol. 12, 2003, pp. 41-58.
[34] P. Vassiliadis, M. Bouzeghoub, and C. Quix, "Towards Quality-Oriented Data Warehouse Usage and Evolution," *LNCS 1626*, 1999.
[35] Y. Cui, J. Widom, and J. L. Wiener, "Tracing the lineage of view data in a warehousing environment," in *ACM Transactions of Database Systems*, vol. 25, 2000, pp. 179-227.
[36] S. Patnaik, M. Meier, B. Henderson, J. Hickman, and B. Panda, "Improving the Performance of Lineage Tracing in Data Warehouse," *SAC*, 1999.
[37] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita, "Improving Data Cleaning Quality Using a Data Lineage Facility," in *DMDW*, 2001, pp. 3.
[38] P. Buneman, S. Khanna, and W. C. Tan, "On Propagation of Deletions and Annotations Through Views," in *PODS*, 2002, pp. 150-158.
[39] D. Gannon, R. Bramley, G. Fox, S. Smallen, Al~Rossi, R. Ananthakrishnan, F. Bertrand, K. Chiu, M. Farrellee, M. Govindaraju, S. Krishnan, L. Ramakrishnan, Y. Simmhan, A. Slominski, Yu~Ma, C. Olariu, and N. Rey-Cenvaz, "Programming the Grid: Distributed Software Components, P2P and Grid Web Services for Scientific Applications," in *Cluster Computing*, vol. 5, 2002, pp. 325-336.
[40] "The Grid: Blueprint for a New Computing Infrastructure," I. Foster and C. Kesselman, Eds., 1998.
[41] F. Leymann, "Web Services Flow Language (WSFL 1.0)," *IBM Software Group Technical Report*, 2001.
[42] "Business Process Execution Language for Web Services (BPEL 1.1)," *Technical Report: BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems*, 2003.
[43] B. Plale, D. Gannon, D. Reed, S. Graves, K. Droegemeier, B. Wilhelmson, and M. Ramamurthy, "Towards Dynamically Adaptive Weather Analysis and Forecasting in LEAD," in *ICCS workshop on Dynamic Data Driven Applications*, 2005.
[44] P. Groth, S. Miles, W. Fang, S. C. Wong, K.-P. Zauner, and L. Moreau, "Recording and Using Provenance in a Protein Compressibility Experiment," *HPDC*, 2005.
[45] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," in *SSDBM*, 2002, pp. 37-46.
[46] J. Zhao, C. Wroe, C. A. Goble, R. Stevens, D. Quan, and R. M. Greenwood, "Using Semantic Web Technologies for Representing E-science Provenance," in *International Semantic Web Conference*, 2004, pp. 92-106.
[47] J. D. Myers, T. C. Allison, S. Bittner, B. T. Didier, M. Frenklach, W. H.~Green, Jr., Y.-L. Ho, J. C. Hewson, W. S. Koegler, C. Lansing, D. Leahy, M. Lee, R. McCoy, M. Minkoff, S. Nijsure, G. v. Laszewski, D. Montoya, C. M. Pancerella, R. Pinzon, W. Pitz, L. A. Rahn, B. Ruscic, K. Schuchardt, E. Stephan, Al~Wagner, T. L. Windus, and C. L. Yang, "A Collaborative Informatics Infrastructure for Multi-scale Science," in *CLADE*, 2004, pp. 24.
[48] H.-I. Hsiao and I. Narang, "DLFM: A Transactional Resource Manager," *PODS*, 2000.
[49] L. Chiticariu, W.-C. Tan, and G. Vijayvargiya, "DBNotes: A Post-It System for Relational Databases based on Provenance," *SIGMOD*, 2005.
[50] D. Bhagwat, L. Chiticariu, W. C. Tan, and G. Vijayvargiya, "An Annotation Management System for Relational Databases," in *VLDB*, 2004, pp. 900-911.
[51] Y. Cui and J. Widom, "Practical Lineage Tracing in Data Warehouses," in *ICDE*, 2000, pp. 367-378.
[52] Y. R. Wang and S. E. Madnick, "A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective," in *VLDB*, D. McLeod, R. Sacks-Davis, and H.-J. Schek, Eds., 1990, pp. 519-538.
[53] A. Woodruff and M. Stonebraker, "Supporting Fine-grained Data Lineage in a Database Visualization Environment," in *ICDE*, 1997, pp. 91--102.
[54] J. Widom, "Trio: A System for Integrated Management of Data, Accuracy, and Lineage," in *CIDR*, 2005, pp. 262-276.

[55] D. P. Lanter, "Lineage in GIS: The Problem and a Solution," in *Technical Report*: National Center for Geographic Information and Analysis, 1990.
[56] J. L. Wason, M. Molinari, Z. Jiao, and S. J. Cox, "Delivering Data Management for Engineers on the Grid," in *Euro-Par*, 2003, pp. 412-416.
[57] S. Krishnan, R. Bramley, D. Gannon, R. Ananthakrishnan, M. Govindaraju, A. Slominski, Y. Simmhan, J. Alameda, R. Alkire, T. Drews, and E. Webb, "The XCAT Science Portal," in *Scientific Programming*, vol. 10, 2002, pp. 303-317.
[58] R. K. Bose, "Composing and Conveying Lineage Metadata for Environmental Science Research Computing," in *Ph.D. Thesis, University of California, Santa Barbara*, 2004.
[59] D. Pearson, "Presentation on Grid Data Requirements Scoping Metadata & Provenance," in *Workshop on Data Derivation and Provenance, Chicago*, 2002.
[60] G. Cameron, "Provenance and Pragmatics," in *Workshop on Data Provenance and Annotation, Edinburgh*, 2003.
[61] C. Goble, "Position Statement: Musings on Provenance, Workflow and (Semantic Web) Annotations for Bioinformatics," in *Workshop on Data Derivation and Provenance, Chicago*, 2002.
[62] P. P. da~Silva, D. L. McGuinness, and R. McCool, "Knowledge Provenance Infrastructure," in *IEEE Data Engineering Bulletin*, vol. 26, 2003, pp. 26-32.
[63] S. Miles, P. Groth, M. Branco, and L. Moreau, "The requirements of recording and using provenance in e-Science experiments," in *Technical Report, Electronics and Computer Science, University of Southampton*, 2005.
[64] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao, "The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration," in *CIDR*, 2003.
[65] H.~Veregin and D. P. Lanter, "Data-quality enhancement techniques in layer-based geographic information systems," in *Computers, Environment and Urban Systems*, vol. 19, 1995, pp. 23-36.
[66] H. Müller and F. Naumann, "Data Quality in Genome Databases," in *IQ*, 2003, pp. 269-284.
[67] M. S. E. Burgess, W. A. Gray, and N. J. Fiddian, "A Flexible Quality Framework for Use within Information Retrieval," in *International Conference on Information Quality*, 2003, pp. 297-313.
[68] M. Barth, P. Miller, and A. MacDonald, "MADIS: Providing Value-Added Observations to the Meteorological Community," *FSL Forum White Paper*, 2002.
[69] "Sarbanes-Oxley Act," *Securities and Exchange Commission, USA, http://www.sec.gov*, 2002.
[70] H. Fan and A. Poulovassilis, "Tracing Data Lineage Using Schema Transformation Pathways," in *Knowledge Transformation for the Semantic Web*, 2003, pp. 64-79.
[71] R. Bose, "A Conceptual Framework for Composing and Managing Scientific Data Lineage," in *SSDBM*, 2002, pp. 15-19.
[72] R. Williams, J. Bunn, R. Moore, J. C. T. Pool, and Cacr, "Interfaces to Scientific Data Archives," in *Report on NSF Workshop*: Center for Advanced Computing Research, California Institute of Technology, 1998.
[73] J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens, "Annotating, linking and browsing provenance logs for e-Science," in *ISWC Workshop on Retrieval of Scientific Data, Florida*, 2003.
[74] J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer, "Semantically Linking and Browsing Provenance Logs for E-science," in *ICSNW*, 2004, pp. 158-176.
[75] B. Howe and D. Maier, "Modeling Data Product Generation," in *Workshop on Data Derivation and Provenance, Chicago*, 2002.
[76] A. P. Marathe, "Tracing Lineage of Array Data," in *SSDBM*, 2001, pp. 69-78.
[77] B. Plale, J. Alameda, B. Wilhelmson, D. Gannon, S. Hampton, A. Rossi, and K. Droegemeier, "Active Management of Scientific Data," *Internet Computing*, 2005.
[78] Ncsa, "HDF5, http://hdf.ncsa.uiuc.edu/HDF5," 2005.
[79] J. Myers, C. Pancerella, C. Lansing, K. Schuchardt, and B. Didier, "Multi-Scale Science, Supporting Emerging Practice with Semantically Derived Provenance," in *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, 2003.
[80] R. Bose and J. Frew, "Composing Lineage Metadata with XML for Custom Satellite-Derived Data Products," in *SSDBM*, 2004, pp. 275-.
[81] R. Fileto, C. B. Medeiros, L. Liu, C. Pu, and E. D. Assad, "Using Domain Ontologies to Help Track Data Provenance," in *Brazilian Symposium on Databases (SSDB)*, vol. 18, 2003, pp. 84-98.
[82] Nost, "Definition of the Flexible Image Transport System (FITS)," NASA/Science Office of Standards and Technology, 1999.

[83] P. Buneman, S. Khanna, K. Tajima, and W.-C. Tan, "Archiving Scientific Data," in *SIGMOD*, 2002, pp. 1--12.
[84] P. Groth, M. Luck, and L. Moreau, "A protocol for recording provenance in service-oriented Grids," in *OPODIS*. Grenoble, France, 2004.
[85] D. P. Lanter, "A Lineage Metadata Approach to Removing Redundancy and Propagating Updates in a GIS Database," in *Cartography and Geographic Information Systems*, vol. 21, 1994, pp. 91-98.
[86] D. P. Lanter, "A Lineage Meta-Database Approach Towards Spatial Analytic Database Optimization," in *Cartography and Geographic Information Systems*, vol. 20, 1993, pp. 112-121.
[87] R. D. Stevens, A. J. Robinson, and C. A. Goble, "myGrid: personalised bioinformatics on the information grid," in *Bioinformatics*, vol. 19, 2003, pp. 302i-304.
[88] C. Pancerella, J. Hewson, W. Koegler, D. Leahy, M. Lee, L. Rahn, C. Yang, J. D. Myers, B. Didier, R. McCoy, K. Schuchardt, E. Stephan, T. Windus, K. Amin, S. Bittner, C. Lansing, M. Minkoff, S. Nijsure, G. v. Laszewski, R. Pinzon, B. Ruscic, Al~Wagner, B. Wang, W. Pitz, Y.-L. Ho, D. Montoya, L. Xu, T. C. Allison, W. H. Green, Jr, and M. Frenklach, "Metadata in the collaboratory for multi-scale chemical science," in *Dublin Core Conference*, 2003.
[89] C. Pancerella, J. Myers, and L. Rahn, "Data Provenance in CMCS," in *Workshop on Data Derivation and Provenance, Chicago*, 2002.
[90] "Dublin Core Metadata Initiative," *http://www.dublincore.org*, 2005.
[91] M. Szomszor and L. Moreau, "Recording and Reasoning over Data Provenance in Web and Grid Services," in *ODBASE*, 2003, pp. 603-620.
[92] J. Frew and R. Bose, "Earth System Science Workbench: A Data Management Infrastructure for Earth Science Products," in *SSDBM*, 2001, pp. 180-189.
[93] A. Aiken, J. Chen, M. Stonebraker, and A. Woodruff, "Tioga-2: A Direct Manipulation Database Visualization Environment," *ICDE*, 1996.
[94] P. Buneman, S. Khanna, and W. C. Tan, "Data Provenance: Some Basic Issues," in *FSTTCS*, 2000, pp. 87-93.
[95] P. Buneman, S. Khanna, and W. C. Tan, "Data Archiving," in *Workshop on Data Derivation and Provenance, Chicago*, 2002.
[96] Y. Cui, J. Widom, and J. L. Wiener, "Tracing the Lineage of View Data in a Data Warehousing Environment," in *Technical Report*: University of California, 1997.
[97] T. Berners-Lee, J. Hendler, and O. Lassila, " The Semantic Web," *Scientific American*, 2001.
[98] C. A. Lynch, "When documents deceive: Trust and provenance as new factors for information retrieval in a tangled web," in *JASIST*, vol. 52, 2001, pp. 12-17.
[99] G. Miklau and D. Suciu, "Enabling Secure Data Exchange," in *Data Engineering Bulletin, Special Issue on Data Security and Privacy*, 2004.