Lineage Retrieval for Scientific Data Processing: A Survey

RAJENDRA BOSE AND JAMES FREW

Bren School of Environmental Science and Management
University of California, Santa Barbara

Scientific research relies as much on the dissemination and exchange of data sets as on the publication of conclusions. Accurately tracking the lineage (origin and subsequent processing history) of scientific data sets is thus imperative for the complete documentation of scientific work. Researchers are effectively prevented from determining, preserving, or providing the lineage of the computational data products they use and create, however, because of the lack of a definitive model for lineage retrieval and a poor fit between current data management tools and scientific software. Based on a comprehensive survey of lineage research and previous prototypes, we present a metamodel to help identify and assess the basic components of systems that provide lineage retrieval for scientific data products.

Categories and Subject Descriptors: J.2 [**Computer Applications**]: Physical Sciences and Engineering; J.3 [**Computer Applications**]: Life and Medical Sciences—*Biology and genetics*; H.3.m [**Information Storage and Retrieval**]: Miscellaneous; H.4.2 [**Information System Applications**]: Types of Systems; K.6.4 [**Management of Computing and Information Systems**]: System Management—*Management audit*

General Terms: Design, Documentation, Experimentation, Management

Additional Key Words and Phrases: Data lineage, data provenance, scientific data, scientific workflow, audit

1. INTRODUCTION

Scientific investigation often relies as much on the broad dissemination and exchange of data sets as on the publication of conclusions. Whether by further processing within the originating organization or propagation to other research groups, a particular assemblage of data may contribute to other derivative data products over time. One common example of this is the chain (or pipeline) of processing steps used to generate lower levels of National Aeronautics and Space Administration (NASA) remote sensing data products (Table I). As scientists become online data providers, the availability and transmission of various levels of digital

This research was supported by NASA Cooperative Agreement NCC5-302.
Authors' addresses: R. Bose, School of Informatics, University of Edinburgh, Appleton Tower, Crichton Street, Edinburgh EH8 9LE; email: rbose@inf.ed.ac.uk; J. Frew, Bren School of Environmental Science and Management, Bren Hall, University of California, Santa Barbara, CA 93106-5131; email: frew@bren.ucsb.edu.

Table I.

Data Level	Description
Level 0	Reconstructed unprocessed instrument data at full resolutions.
Level 1A	Reconstructed, unprocessed instrument data at full resolution, time referenced, and annotated with ancillary information, including radiometric and geometric calibration coefficients and georeferencing parameters (i.e., platform ephemeris) computed and appended, but not applied to the Level 0 data.
Level 1B	Level 1A data that has been processed to sensor units (i.e., radar backscatter cross section, brightness temperature, etc.). Not all instruments will have a Level 1B equivalent.
Level 2	Derived environmental variables (e.g., ocean wave height, soil moisture, ice concentration) at the same resolution and location as the Level 1 source data.
Level 3	Variables mapped on uniform space-time grid scales, usually with some completeness and consistency properties (e.g., missing points interpolated, complete regions mosaicked together from multiple orbits.
Level 4	Model output or results from analyses of lower-level data (i.e., variables that were not measured by the instruments but instead are derived from these measurements.

(From National Aeronautics and Space Administration (NASA) [1986]).

Table II.

Retrieve Lineage of:	Example Domains	References
Geographic information system (GIS) data layers	environmental impact on land parcels; temporal records for property boundaries	[Lanter 1991; Lanter 1993; Spery et al. 1999]
constituent items in computational flow for scientific models or simulations	hydrologic and climate models; satellite data processing	[Alonso et al. 1993; Smith et al. 1993; Alonso et al. 1997b; Woodruff et al. 1997; Frew et al. 2001]
query results for DBMS, XML, data warehouse	research using molecular biology databases; commercial applications	[Buneman et al. 2000a; Cui et al. 2000; Buneman et al. 2001; Buneman et al. 2002b; Cui et al. 2003]
results of web or Grid service requests	physics experiments; biology e-Science investigations	[Foster et al. 2003; Zhao et al. 2003]
results of scientific laboratory work	multi-scale chemical science	[Myers et al. 2003a; Myers et al. 2003b]
data analysis session in interactive programming environment	biology: analyzing species' physical traits	[Becker et al. 1988]
operating system processes and files	C programming, web applications	[Vahdat et al. 1998]
environmental data	environmental impact on land parcels	[Eagan et al. 1993]

data products over computer networks only complicates the possible connections between related data sets and processing algorithms.

During the past two decades, research projects have yielded designs and prototypes for computing and information systems that preserve and retrieve the origins and processing history—that is, the *lineage*—of objects and processes. Table II summarizes the objects of lineage retrieval and example domains for the major areas of interest.

Geographic metadata standards from the last decade address the issue of spatial data transfer between disparate groups and systems by requiring lineage as part of a data quality report. The data quality information is included to protect potential data consumers from unintended consequences resulting from, for example, misinformation or mistaken assumptions about data collection methods, measurement precision, or scale. Most research projects involving lineage are motivated by the benefits that providers of scientific

Table III.

Data Quality Benefits of Lineage	References
Communicates data quality: suitability, reliability, accuracy, currency, redundancy.	[Lanter 1991; Eagan et al. 1993; Clarke et al. 1995; Buneman et al. 2000b]
Enhances interpretation, prevents misinterpretation, misuse of environmental data.	[Eagan et al. 1993]
Enhances a user's justification for using data.	[Eagan et al. 1993]
Reduces possible false sense of data precision.	[Eagan et al. 1993]
Facilitates integration of data for regional analysis.	[Eagan et al. 1993]
Allows nonexpert data user to understand processing steps.	[Woodruff et al. 1997]
Communicates processing steps leading to creation of scientific data product.	[Brown et al. 1995; Buneman et al. 2000b; Frew et al. 2001]
Allows access to sources of materialized relational views; "drill down."	[Cui et al. 1997; Buneman et al. 2001]
Allows updates to sources from materialized relational views.	[Cui et al. 1997]
Allows modification of materialized relational view schema.	[Cui et al. 1997]
Enables future generations to use historical data resources.	[Clarke et al. 1995]
Documents geographical changes from successive updates to a reference cadastral DBMS.	[Spery et al. 1999]

data, and other future users, receive from the ability to track the lineage of computational results such as accounting for errors or knowing how algorithms have been combined. These and other examples of the benefits of lineage that support the goal of supplying data quality are included in Table III, and examples of the benefits of lineage that support the management of scientific processing are included in Table IV.

1.1. Definition of Terms

There is no standard terminology for referring to data processing activities in the sciences; different research groups and communities are almost certain to use different vocabularies. To prevent confusion, the terms used in this article are defined here.

This survey focuses on computer-based data processing with limited human intervention rather than mostly interactive activities such as manipulating data in a spreadsheet. We recognize several loosely defined, and potentially overlapping, categories of data processing: script- or program-based, query-based, workflow management system (WFMS)-based, and service-based.

Script-based data processing is usually performed with dynamically-typed, general-purpose interpreted languages such as Python and Perl, or with the languages in so called problem solving environments such as the commercial products Interactive Data Language (IDL) [Research Systems Inc. 2003] and MATLAB [Mathworks 2003], where a primary objective is the rapid development and immediate execution of code. *Program-based processing* refers to the use of statically-typed, compiled languages such as C/C++, Fortran, and Java.

The remaining data processing categories may also use or require script-like constructs but are generally subject to more constraints than imposed by the previously defined scripting environments. *WFMS-based data processing* requires instructions expressed in a specific process-definition language and the registration or wrapping of external code. *Query-based processing* relies on submitting queries to a database management system (DBMS), and *service-based processing* relies on a network of web servers or Grid [Foster

Table IV.

Scientific Processing Benefits of Lineage	References
Records processing history for internal records, audit, quality control.	[Clarke et al. 1995; Frew et al. 2001]
Records computational history for judging statistical validity of future operations.	[Buneman et al. 2000b]
Reduces data provider liability.	[Eagan et al. 1993]
Provides consistent documentation for distributed data sets.	[Eagan et al. 1993]
Finds the sources of faulty, anomalous processing outputs.	[Brown et al. 1995; Cui et al. 1997; Woodruff et al. 1997; Buneman et al. 2000b]
Finds the outputs affected by faulty, anomalous processing inputs.	[Brown et al. 1995; Woodruff et al. 1997; Buneman et al. 2000b]
Saves processing "recipes"; modify and rerun processing sequence.	[Woodruff et al. 1997; Buneman et al. 2000b]
Optimizes spatial database size by creating interim products on demand.	[Lanter 1989b; Lanter 1993]
Compares the analytical steps of two or more GIS applications.	[Lanter 1994]
Assists in propagating error measures during processing.	[Lanter et al. 1990]
Supports object version control for cooperative modeling in scientific DBMS.	[Alonso 1994]
Allows monitoring system to identify source of faulty data collected over network.	[Cui et al. 1997]
Allows an information center to notify data sources after "data cleansing" operations.	[Cui et al. 1997]

et al. 1999] nodes. In Section 3, lineage-related research for these various categories of data processing is discussed.

When referring to *data processing workflow*, we use the term workflow in the broad sense of a sequence of tasks, without implying the use of specialized software such as a WFMS or workflow engine. Workflow and lineage are related, but subtly different notions. Workflow is prospective in nature and defines plans for desired processing. Lineage, on the other hand, is retrospective like an audit [Becker and Chambers 1988] and describes the relationships between data products and data transformations after processing has occurred. Workflow is discussed further in Section 3.

As genealogical charts reveal successive generations of parents for an individual, the lineage of an item describes how it was derived from its source. The lineage of a data product refers to its sources and derivation [Clarke and Clark 1995; Woodruff and Stonebraker 1997], or as summarized by Eagan and Ventura [1993]: "all the processes and transformations of data from original measurements to current form." Thus, in addition to source observations or materials, the lineage of a data product encompasses data acquisition and compilation methods, conversions, transformations, and analyses, along with the assumptions and criteria applied at any stage of the data product life cycle [Clarke and Clark 1995].

Lineage may also apply to items that have evolved from a data product. Two forms of navigating lineage are thus implied: moving backward to discover ancestor products or transformations, or moving forward to discover descendant products or transformations.

The terms *data provenance* and *data pedigree* [French 1995; Buneman et al. 2000b] have been used interchangeably to refer to the sources of query- and service-based data processing results, while lineage connotes the processing history of a data product. Derivation history [Hachem et al. 1993], data set dependence [Alonso et al. 1998], filiation [Spery et al. 1999], data genealogy [Barkstrom 1998], data archeology, and audit trail [Brown and Stonebraker 1995] are other related terms used in the literature.

1.2. The Problem of Irretrievable Lineage

Scientific researchers face the problem that, although they have growing responsibilities as online data providers [National Research Council 1999], and they generate data that contribute to the scientific archive, the lineage of the data products they create is often irretrievable. Ideally, after processing has occurred, the documentation for workflow invocations should at least provide the ability to retrieve and understand the relationships between data products and the scripts or programs that used or generated them.

This remains a challenging task and is not usually achieved for two reasons. First, tools for composing lineage metadata are not provided with the software used for much of scientific data processing. The ability of researchers to generate data in contemporary computing environments can quickly exceed their ability to track how it was created [Lanter 1990]. Ironically, recording the processes used to create data products has become more difficult and tedious as computational tools have become more sophisticated [Clarke and Clark 1995]. Second, no definitive method, standard, or mandate exists for preserving, providing, or communicating the lineage of computational results.

1.3. Describing a Metamodel for Lineage Retrieval

One goal for this survey is to arrive at a general model for systems that allow the retrieval of lineage for processing results. After mentioning standards relevant to lineage, the review of lineage research in Section 3 is organized according to the mode of data processing that creates the target items for lineage retrieval. Other types of computer support for scientific work, including the broadly defined areas of experiment, workflow, and version management, complement the objective of lineage retrieval, especially in designing systems for scientific data processing. Sections 4–6 cover research for these related research areas. Following this, a metamodel for systems that include lineage is introduced where lineage retrieval is dependent on the workflow and metadata models designed into the systems. Four recent systems that share the goal of tracking and retrieving lineage for new scientific data products are discussed within the context of the metamodel. Finally, trends in methods for lineage retrieval are summarized.

2. LINEAGE-RELATED STANDARDS

Federal metadata standards for geospatial data have included specifications for lineage as a component of data quality information since the Spatial Data Transfer Standard (SDTS) [U.S. Geological Survey 1992] became a Federal Information Processing Standard (FIPS 173) in 1992. The SDTS, developed for transferring georeferenced spatial data between dissimilar applications or computer systems, includes a brief text description of lineage as part of a data quality report. This report is required not only to accompany the data in a standard transfer, but also to be obtainable separately from the actual data (No mechanism is given for how this is to be achieved). The SDTS influenced the Federal Geographic Data Committee (FGDC) Content Standard for Digital Geospatial Metadata (CSDGM) [Federal Geographic Data Committee 1998] which defines explicit metadata elements corresponding to digital geospatial data sets, including lineage. CSDGM lineage comprises two compound metadata elements, source information and process step, with each of these two main elements repeated as necessary. Neither of these standards specifies "the means by which this [lineage] information is organized in a computer system or in a data transfer, nor the means by which this information is transmitted, communicated, or presented to the user" [Federal Geographic Data Committee 1998]. The CSDGM, however, is the most authoritative model of geospatial data lineage to date. Figure 1 provides an example of CSDGM data lineage elements.

Eagan and Ventura [1993] consider the issues involved with using lineage

Lineage:
Source_Information:
Source_Citation:
Citation_Information:
Originator: University of Miami/Rosenstiel School of Marine and Atmospheric Sciences
Publication_Date: 1991
Title: NOAA Advanced Very High Resolution Radiometer Multichannel Sea Surface Temperature data set
Geospatial_Data_Presentation_Form: remote-sensing image
Other_Citation_Details: Distributed by the Distributed Active Archive Center, Jet Propulsion Laboratory, Pasadena, California. User's guide is 10 pages. The data are distributed on 9 nine-track tapes in VAX Backup format.
Source_Citation_Abbreviation: AVHRR MCSST
Type_of_Source_Media: Nine-track tape
Source_Time_Period_of_Content:
Time_Period_Information:
Range_of_Dates/Times:
Beginning_Date: 19811001
Ending_Date: 19891231
Source_Currentness_Reference: Smith, E., 1991, A user's guide to the NOAA Advanced Very High Resolution Radiometer Multichannel Sea Surface Temperature data set produced by the University of Miami/Rosenstiel School of Marine and Atmospheric Science: Distributed by the Distributed Active Archive Center, Jet Propulsion Laboratory, Pasadena, California. 10 p.
Source_Contribution: The source data set provides 467 weekly images of each of nine regions of the world oceans; these weekly files were averaged in the present data set to produce monthly composite images.
Process_Step:
Process_Description: Calculate monthly averages and composite monthly averages. The included C-language programs sum.c and combine.c were used to calculate the monthly and weekly average sea surface temperature files. For each grid cell in the images, sum.c calculates the arithmetic average of the corresponding cell in the input files for each month or week of the year. Results are written to an set of intermediate files which are interpreted by combine.c. The program combine decodes the intermediate files written by sum and writes each average image into a new file.
Process_Date: 1993
Process_Step:
Process_Description: Create GIF and PICT images of monthly and weekly averages. The C-language program mrletoppm.c converts a monthly or weekly average file into a portable pixmap. GIF and PICT images were derived from these pixmaps using the freely available PbmPlus toolkit developed by Jeff Poskanzer
Process_Date: 1993

Fig. 1. Example of FGDC CSDGM data lineage (from U.S. Geological Survey [1995]).

to improve documentation for transferred data, including nonspatial scientific data. They suggest using data lineage reports to notify downstream data users of the limitations and original intent of environmental data sets. Their example environmental data lineage report includes thorough data source and transformation descriptions similar to the lineage elements in an SDTS quality report. They also suggest that data lineage reports can help address issues of distribution, accessibility, reliability, and currency for environmental data.

3. LINEAGE RETRIEVAL FOR DATA PROCESSING SYSTEMS

Section 1.1 mentions different categories or modes of data processing that create the objects of lineage retrieval: script- and program-based, WFMS-based, query-based, and service-based data processing. To these four categories we add *command*

line-driven data processing which is the focus of early lineage research.

The key distinction between these categories is the locus of data processing control, or what drives data transformations. Command line-, script- and program-based processing suggest the enactment of one or more single-user processing threads at the operating system or scripting environment level. WFMS-based processing is controlled by a workflow engine, while query-based processing is dependent on the functioning of one or more DBMSs. Finally, web or Grid service-based processing relies on a network of web servers or Grid nodes.

As discussed earlier, these categories are somewhat artificial; a realistic data processing example might include aspects of many of these categories. For example, in Zhao et al. [2003], web service requests are enacted by scripts submitted to a WFMS. However, this system falls under the category of service-based data processing because it is the web service that ultimately applies transformations to input items to create new output items.

3.1. Command Line-Based Data Processing

The systems described in this section are implemented by monitoring a command line interpreter which allows them to passively capture and store the information necessary to assemble a retrospective view on data processing.

As defined by Merriam-Webster [2001], an *audit trail* is a record of a sequence of events (as actions performed by a computer) from which a history can be reconstructed, and thus serves as a form of lineage. Becker and Chambers [1988] describe a system for auditing data analyses steps for a particular implementation of S, a language and interactive environment for statistical analysis and display. Their intention is to provide a tool for a user to investigate the dependencies among steps following an exploratory S analysis session. User-entered statements evaluated by S, including the associated creation and modification of data objects resulting from those statements, are dynamically recorded in an audit file. An audit facility parses the audit file into a linked list structure, which it then uses to respond to ad hoc queries and generate custom so-called *audit plots* to display analysis step dependencies. The prototype audit facility Becker and Chambers describe has not been implemented in contemporary versions of the S system such as S-Plus [Insightful Corporation 2003].

In the early 1990s, Lanter contributed a body of work centered on designing a metadatabase to track the lineage of operations within a geographic information system (GIS) [1988; 1989a,b; 1990; 1991; 1993; 1994; Lanter and Veregin 1990]. He explores using data lineage to optimize the size of spatial databases [Lanter 1989b; 1993], compare spatial analytic GIS applications [Lanter 1994], and propagate measures of error through GIS applications [Lanter and Veregin 1990].

Lanter and Veregin's Lineage Information Program (LIP) [1990], which later evolved into a commercial product called Geolineus, was built on the ARC/INFO [ESRI 1982] GIS and provided lineage strictly for ARC/INFO operations. By monitoring user input at the command line, the LIP provided instructions prompting the user to enter metadata in response to particular ARC/INFO commands and allowed simple lineage queries with responses to be delivered to the screen. Although no longer available, Geolineus remains one of the few operational (as opposed to prototype) lineage tracking systems cited in the literature.

Vahdat and Anderson [1998], in an approach similar in concept to Lanter's work, describe an implementation of their Transparent Result Caching (TREC) prototype at the level of a UNIX programming shell. By intercepting read/write system calls from a shell, TREC builds process and file dependency information and caches the results. The cached process lineage can be queried and exploited for several practical tasks, including keeping web page caches current and providing an "unmake" utility to navigate backwards through the captured lineage. While a "make" file [Feldman 1978] provides the

means to explicitly define file dependencies, Vahdat and Anderson's unmake function queries the transparently (that is, automatically) cached dependency graph to specify the sequence of processes and files used to create a particular output file. Unmake can thus be used to retroactively create a make file.

3.2. Script- and Program-Based Data Processing

The systems described in this section assemble a retrospective view on processing using information encoded directly in user-supplied scripts or programs.

ESSW [Frew and Bose 2001] captures lineage metadata for objects involved in scientific processing performed with application-specific scripts as well as general scripting languages such as Perl. ESSW uses custom application programming interface (API) commands within Perl *wrapper scripts*—code that circumscribes the functions, algorithms, or other data transformations of interest—to construct lineage. The lineage of an item is queried through a web application, and results are displayed diagrammatically using the Webdot Web service interface included with the Graphviz set of graphing tools [AT&T 2001].

A Semantic Web-related project, Geodise [Chen et al. 2003], is similar to ESSW in that it is concerned with issues of metadata creation and workflow with a widely-used scripting environment (in their case, MATLAB). They provide a standalone workflow editor application to create scripts and provide additional functionality, using the Java interface to MATLAB. Geodise, however, is not designed to track the lineage of items created by the execution of scripts.

Marathe [2001] provides an algorithm to compute fine-grain data lineage (see discussion in Section 3.4) but only for array operations expressed in the Array Manipulation Language (AML). He performs lineage tracing by systematically applying a set of rewrite rules to the original AML expression, or operator tree, for an array subset of interest.

GOOSE [Alonso and El Abbadi 1993], a prototype system associated with the multidisciplinary Amazon research project [Smith et al. 1993], uses *data object* attributes to store pointers to the original and latest versions of any inputs and outputs. With this information, a *cooperation graph* of objects and transformations can be constructed to both track object versions and trace lineage in a graphical interface. Transformations in GOOSE are scripts that call external C or Fortran programs. GOOSE is a high-level support environment for cooperative modeling that requires all transformations and data items to be entered or registered as objects internal to GOOSE prior to performing data processing.

3.3. WFMS-Based Data Processing

Extending some of the concepts in GOOSE, the Geo-Opera extension of the OPERA kernel [Alonso and Hagen 1997b; Alonso et al. 1998] provides a management system for distributed geoprocessing that incorporates elements of workflow management, transaction processing, and lineage tracking for an Earth Science example of hydrologic modeling. Data files and transformations used by hydrologic models reside outside of the system. Once transformations are registered in Geo-Opera, they are tracked as *task objects* internal to the system. Lineage relationships between objects are established by defining the control flow between internal task objects and data. When data is located outside the system, it is registered in the system as an *external object*. Each external object includes a set of system-maintained attributes supporting automated versioning, change propagation, and lineage recording.

3.4. Query-Based Data Processing

Brown and Stonebraker [1995] and Woodruff and Stonebraker [1997] propose a method for providing detailed or fine-grained lineage for scientific processing applications. A goal of their research

is delivering to scientists, through data lineage, the ability to investigate the source of faulty or anomalous data sets and the ability to determine those derived data sets affected by faulty or anomalous inputs or algorithms. Specifically, Woodruff and Stonebraker [1997] address the problem of recovering the origins of single elements in large arrays of data that have undergone a series of transformations. Creating individual metadata entries to assist with such a task would require prohibitive effort and storage size.

Their method requires that all processing be performed within a DBMS. Fine-grained data lineage is accessible only when all user-defined processing algorithms, plus special additional functions for each algorithm, have been registered and stored in Tioga [Stonebraker et al. 1993], a database visualization environment built over the POSTGRES DBMS [UC Berkeley 1994]. The additional functions supplied are weak inversion and verification functions that are used to resolve lineage queries on the fly. Because not all functions or user-defined algorithms are perfectly invertible, a *weak inversion function* is meant to provide an imperfect but still useful mapping into database element inputs that may be responsible for a given output. For those functions that cannot be inverted without reference to input values, a *verification function* with access to input values for the original function is defined to further refine the mapping.

Buneman et al. [2000b] assess the limitations of current DBMS technology in providing annotations and provenance for shared scientific data. They consider a source database, possibly curated (managed by a human expert), that is queried to provide data values to some target database. The authors contend that because the source is unaware of the target(s) it supplies, it cannot notify targets when updates to source data occur. They also make the point that many target databases are likewise unaware of their sources, thus inverse queries are not possible. In their view, both source and target databases lack the ability to track change histories, support annotations of variable granularity, and communicate details implicit across all records. The authors present a set of open research issues related to providing data annotation and provenance to application domains, such as molecular biology, linguistics, and ecology, that have a history of DBMS use. They suggest the need to move toward active coordination of source and target database interaction.

Several papers begin to address the questions posed in Buneman et al. [2000b]. Cui et al. [2000] investigate the lineage of views materialized from several different source DBMSs in a data warehouse system. Their work centers on developing formal algorithms to perform lineage tracing for relational views at the tuple level. In Cui and Widom [2003], the authors consider sequences of transformations related to maintaining a data warehouse. This paper defines a set of transformation properties, and develops tracing algorithms that use those properties when specified, to improve the efficiency of lineage tracing. Buneman et al. [2001] differentiate between their definition of "where" provenance that explains where an element in a view came from, and the "why" provenance of Cui et al. [2000] that explains why an element in a view possesses a particular value. Buneman et al. [2001] use a deterministic model of data, where the location of each piece of data has a unique identifying path. With their deterministic model and a query language for semistructured data, they describe and investigate provenance for relational queries and views.

Because Cui et al. [2000], Buneman et al. [2001], and Cui and Widom [2003] focus on lineage for queries expressed with relational algebra, their results have limited applicability to scientific processing that is not performed within a DBMS. The techniques of Buneman et al. [2001] may potentially be applied to a deterministic XML mapping—that is, a mapping where the location of each data item expressed in XML can be uniquely described by a path.

3.5. Service-Based Data Processing

The Chimera Virtual Data System (VDS) matches the scope and ambition of the Grid, targeting invocations of data transformations in a "distributed, multi-user, multi-institutional environment" [Foster et al. 2003]. Chimera features a language, the Virtual Data Language (VDL), for defining and manipulating data derivation procedures which are stored in a Virtual Data Catalog (VDC). The VDL serves as a general wrapper for program execution, capable of accommodating Grid request planning. The language is also used to query the VDC to discover or invoke the lineage or pipeline of computations that created a particular data object. Chimera is described as a *virtual data prototype* because it is capable of creating a directed acyclic graph (DAG) of distributed computations that can be submitted to the Grid to regenerate a given data object.

Zhao et al. [2003] compose lineage metadata in the form of "systematic provenance logs," in their case for the "*in silico* experiments of the biological community." As with the examples in Buneman et al. [2000a], annotated databases of others' experimental results feed their computational activities, and as with the Chimera system, their experiments require output from a Grid or Web service. They build provenance based on the logs generated by a workflow engine. They also attempt to capture precise semantic associations between log entries and formal ontologies to create glue metadata for an investigation web of materials related to experiments.

Table V provides a summary of the prototype systems mentioned in Section 3.

4. RELATED RESEARCH: COLLABORATIVE ENVIRONMENTS AND EXPERIMENT MANAGEMENT

Many research projects seek to improve scientific collaboration with computing environments that capture a generic experiment and information life cycle. In addition, experiment- and laboratory-related information systems attempt to streamline the process of conducting computational or other types of experiments.

4.1. Experiment and Information Life Cycle

Descriptions of the general experiment life cycle emphasize the common tasks of experiment design, data acquisition, or retrieval both before and after conducting experiments, and data analysis, exploration, or visualization of experimental results [Chakravarthy et al. 1993; Ioannidis et al. 1993; Medeiros et al. 1995; Frew and Dozier 1997]. We generalize these experiment and information life cycles in Figure 2. To assist in the following discussion, a diagram showing the scope of various types of scientific information systems in the context of the combined cycles is presented in Figure 3. Scripting and programming environments such as IDL and MATLAB assist in conducting experiments as well as performing visualization for analyzing results.

4.2. Collaborative Environments

Several research projects aim to improve scientific collaboration by simplifying researchers' access to computational resources and experimental results over distributed systems. Solutions usually involve modeling data objects and creating and managing metadata. Prototype multiuser systems resulting from projects such as Sequoia 2000 [Stonebraker 1991], Amazonia [Saran et al. 1996], Gaea [Hachem et al. 1993], ZOO [Ioannidis et al. 1996], OPM [Chen and Markowitz 1995b], ESP2Net [Kaestle et al. 1999] and ViNE [Skidmore et al. 1998] attempt to fully encompass the life cycles presented in Figure 2. All of these prototypes introduce architectures for abstract modeling or experiment design using custom data management tools that interface with lower level file systems, DBMSs, or external programs. Many of these systems seek to shield users from low-level data structures by allowing them to create and modify graphical models or experiment schemas.

CRISTAL [Le Goff et al. 1996] is larger and more ambitious than many other

Table V.

System	Years	Description/Goals	Refs.
Chimera	2002-	A virtual data system for representing, querying and automating data derivation	[Foster et al. 2002]
ESSW: Earth System Science Workbench	1998-	A nonintrusive data management infrastructure to record workflow and data lineage for computational experiments.	[Frew and Bose 2001]
Geolineus	1993	A data lineage tracking system for Arc/Info GIS operations.	[Lanter 1991; Geographic Designs 1993]
GOOSE: Geographic Object-Oriented Support Environment	1994	An environment providing modeling capabilities and high level data and model views for an underlying storage system.	[Alonso 1994]
Geo-Opera: Open Process Engine for Reliable Activities	1997-	An environment providing modeling capabilities and high level data and model views for an underlying storage system.	[Alonso 1994]
Tioga; fine grained lineage functionality	1997	A proposal to modify an existing database visualizer built over POSTGRES, where user functions are registered and executed by DBMS, to provide fine grained lineage: lineage is computed from ancillary, user supplied weak inversion and verification functions.	[Woodruff and Stonebraker 1997]
CMCS: Collaboratory for Multi-Scale Chemical Science	2003-	An informatics-based approach to synthesizing multi-scale chemistry information.	[Myers et al. 2003a; Pancerella et al. 2003]
MyGrid	2003-	High-level service-based middleware to support the construction, management and sharing of data-intensive in silico experiments in biology	[Greenwood et al. 2003; Zhao et al. 2003]
S audit facility	1988	An interactive programming environment for data analysis, graphics and numerical computation with auditing capability.	[Becker and Chambers 1988]
TREC: Transparent REsult Caching	1998	A prototype framework for transparently managing process lineage and file dependency information.	[Vahdat and Anderson 1998]

systems because it seeks to track extremely detailed production data for the large number of crystals—manufactured at different geographic locations over the span of several years—used in a high energy physics instrument. Other projects focus on improved data models for scientific experiments or use the model of a shared laboratory notebook.

4.3. Experiment Management

A common data management problem in scientific research is recording and retrieving the details of many related, often tightly coupled, collections of experiments such as those required for sensitivity analyses. Data models for experimentation range from simple black box representations to detailed entity relationship models [Pratt 1995].

Citing the inadequacy of contemporary data models, the Object Protocol Model (OPM) [Chen and Markowitz 1995b] defines classes for *protocol objects* within a commercial DBMS that are designed to track the mix of genome sequencing protocols (methodologies) in a molecular biology laboratory. LabBase [Stein et al. 1994] is a scientific DBMS developed for similar work.

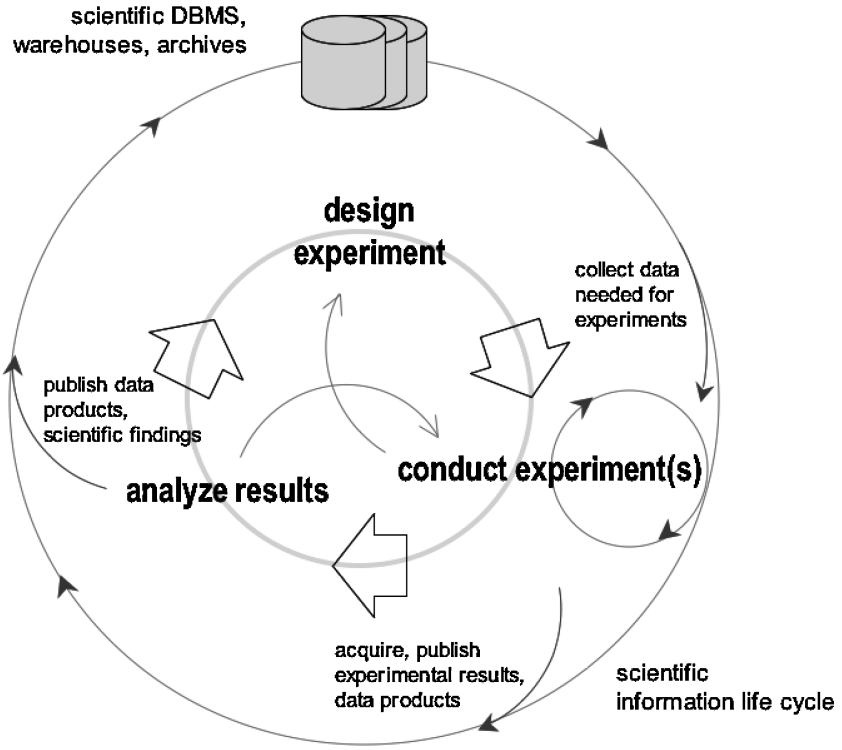

Fig. 2. Combined experiment and information life cycles.

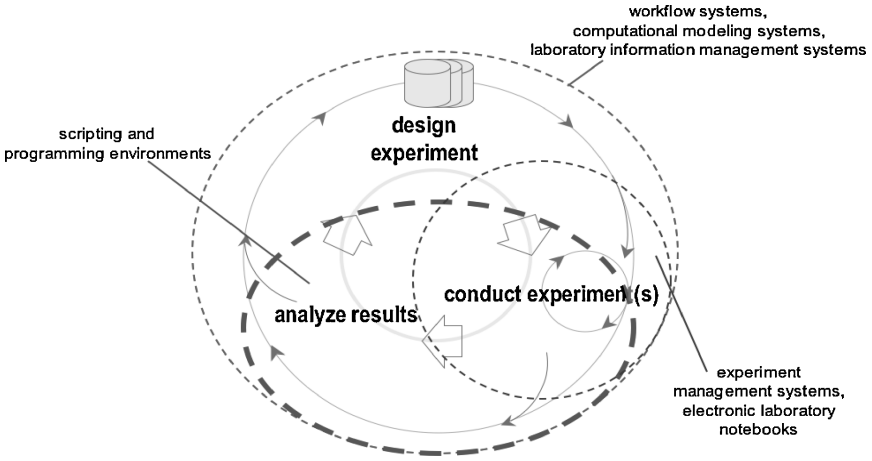

Fig. 3. Type and scope of scientific information systems.

Instead of using dedicated experiment objects as in the OPM collaborative environment, the CCDB project [Cushing et al. 1994] uses the concept of a *proxy object* for computational chemistry experiments in progress. During its lifetime, the proxy object is responsible for generating input files, launching and controlling a computational process over a network until completion, and parsing output files to acquire and store the experiment results. This computational proxy was developed

specifically to address the problems of program and computer platform incompatibility while populating a central database of experiment metadata.

In the ESP^{2}Net system, Scientific Experiment Markup Language (SEML) XML documents are used to "capture the entire experiment experience by including the processes and interrelationships between data and experiments" [Kaestle et al. 1999]. ESSW [Frew and Bose 2001] builds on the data flow concepts of Sequoia 2000 and related work by modeling the flow of scientific experiments with a DAG of unrestricted length, consisting of alternating data and experiment or process step *science objects*. With ESSW, individual researchers are free to choose the level of detail at which to track their experiments but must define a metadata schema for each science object with an XML Document Type Definition (DTD).

The CMCS project [Myers et al. 2003a] supports a flexible approach to provenance suited for collaborations across multiscale chemistry. For CMCS users, provenance refers very broadly to any collection of resources associated with an experiment or study. In their system, the Scientific Annotation Middleware (SAM) provides a WebDAV-capable server [Goland et al. 1999]. When files are entered into the CMCS data repository, XML metadata can be extracted by SAM and assigned to DAV properties in the CMCS schema. SAM can accommodate various views and granularity as well as generate provenance metadata in Resource Description Framework (RDF) [Manola and Miller 2004] form.

4.4. Electronic Laboratory Notebooks

In modern laboratory bench situations, data acquisition or collection may be highly automated with an interface to scientific instruments provided through software. The field of laboratory information management systems has matured over the past two decades, with such systems now able to control instruments distributed over a network and to provide a DBMS repository for input parameters and measurements. Other experimental work, such as in many previously mentioned projects, is largely computational, with computer simulations or other types of scientific data processing performed within commercial data analysis environments (such as IDL, MATLAB, S-Plus, Mathematica (Wolfram Research, Inc.), or Excel (Microsoft Corporation)), or with custom programs in a general programming environment (typically Fortran, C/C++, or Java). Coupled with an appropriate data model and multi-user DBMS, the above systems and environments could assist researchers in reviewing and sharing experimental techniques and results. This is the concept behind an *electronic laboratory notebook*.

Such notebook systems attempt to maintain technical or scientific records in a manner that also satisfies legal and regulatory requirements [Geist and Nachtigal 2003]. For example, standard practice for using paper notebooks in industry and government includes affixing the signature of one or more knowledgeable but disinterested witnesses (known as attestation) to individual pages [Roush 1989]. Attesting carefully documented changes to original work is favored over erasing entries or destroying notebook pages. Attestation is primarily meant to prove the existence of scientific information at a specific point in time, for example, in support of a patent application.

Recent representative electronic notebook projects related to scientific research include the Electronic Lab Notebook, the Virtual Notebook Environment (ViNE), and the Virtual Laboratory Notebook.

The Electronic Lab Notebook [Geist and Nachtigal 2003] emulates the functionality of a paper research notebook in terms of entering basic text and graphics, but also provides the benefits of sharing secure scientific information through a Web browser. The ViNE prototype [Skidmore et al. 1998] provides a more ambitious Web-based notebook interface. Once researchers at a ViNE node define and register the location of their matrix or flat file data and create common gateway interface (CGI) script wrappers for command line computational tools (for example,

MATLAB), they can use an experiment-builder Java applet in a Web browser to create and edit a visual DAG for a set of computational tasks. An equivalent text specification for executing the experiment is automatically generated from the DAG. A Java-based execution controller, capable of multithreaded processing, then runs the experiment from the text specification.

Metadata plays a central role in the prototype Virtual Laboratory Notebook [Winfield 1998], designed to help steer a researcher's interaction with ecological simulation experiments. Once several utility functions have been added to the simulation code, a set of Tcl [Ousterhout 1994] scripts append user annotation, a command log, and a *history tree* as metadata to simulation output data using the Hierarchical Data Format (HDF). A researcher's interactive session with a graphical simulation is captured in a history tree. A researcher's decisions to change simulation parameters are recorded as snapshots, that is, branches in a tree of events. Thus several variations in one simulation are part of one tree which can be viewed through a separate graphical interface. This approach of tying together alternative simulation outcomes is different than the more traditional perspective of considering each simulation as a separate run or workflow.

5. RELATED RESEARCH: WORKFLOW

As an explicit definition of the sequence of activities comprising a process, workflow embodies, but is temporally distinct from, the concept of lineage. Workflow is prospective in nature, while navigating lineage (backward or forward) is only possible after workflow has taken place. Rusinkiewicz and Sheth [1995] offer a useful, succinct definition: "Workflows are activities involving the coordinated execution of multiple tasks performed by different processing entities," be they people or computer programs. Some workflow systems are designed to coordinate activities between a distributed group of collaborating individuals across an enterprise, sometimes referred to as computer-supported collaborative work. This discussion is primarily concerned with more limited systems designed to track data processing for an individual person or a moderate number of concurrent users. This section reviews the application of workflow concepts and WFMS to scientific data processing.

5.1. Process Modeling and Specification

Implementing workflow management requires modeling a workflow process and specifying this model in a form that the WFMS can interpret. A host of process modeling methodologies exist based on DAGs and other types of networks, Petri nets, state charts, or other diagramming tools. No single methodology is dominant, and Alonso et al. [1997a] believe that commercial workflow systems are too dependent on their own idiosyncratic and poorly understood execution models.

The Workflow Management Coalition (WfMC) offers vendors interoperability standards corresponding to their fundamental reference model for WFMS architecture. To facilitate the documentation of process definitions, and their exchange between WFMSs, the WfMC attempts to capture the essence of the different major vendor process definition languages (PDLs) in a standard PDL [Workflow Management Coalition 1999a]. In addition to providing a basic set of diagramming conventions for representing workflow processes [Workflow Management Coalition 1999b], the WfMC has issued an XML definition of its PDL [Workflow Management Coalition 2001]. Their standard PDL is one reference for assessing process modeling and specification approaches for scientific applications. The Working Group for the Process Interchange Format (PIF), another relevant standard, is working with the WfMC to ensure compatibility of their efforts [Lee et al. 1998].

As in business-related enterprises, computing workflows for scientific research may involve passing parameters and data to and from various distributed applications. The notion of Web services promises improvement over

component-based middleware by using XML to wrap underlying computing models [Aoyama et al. 2002]. Although no Web-based workflow system has yet emerged, current activities are developing the components of such an infrastructure. The Business Process Execution Language for Web Services (BPEL4WS) [Thatte 2003] provides a mechanism for encapsulating business processes as Web services, and the Web Services Choreography Description Language (WS-CDL) [Kavantzas et al. 2004] describes how such services can be composed peer-to-peer.

5.2. Scientific Workflow

Comprehensive treatments of WFMS in the literature [Georgakopoulos et al. 1995; Alonso et al. 1997a; Elmagarmid and Du 1997; Mohan 1997; Cichocki et al. 1998; Schael 1998] reflect the business focus of most workflow-related research. Relatively few papers concentrate specifically on incorporating a workflow model into scientific applications. One recent effort in this area is the WASA (Workflow-based Architecture to support Scientific Applications) project at the University of Muenster. WASA is distinguished from the product data management focus of other workflow projects CRISTAL and LabBase by its emphasis on activity descriptions and control [Wainer et al. 1996].

WASA researchers stress the value of moving from data modeling to process modeling. They note that "... workflow systems can prove invaluable in helping activity tracking, data tagging and documentation, even for experiments performed by a single scientist. This is particularly true for scientists working on computational models; they generate large amounts of data, each produced by changing different parameters in the computer models, that must be properly identified" [Wainer et al. 1996]. WASA project work on DNA sequencing and geoprocessing workflow suggest that some scientific applications need only partial specifications for workflow, as opposed to the fully compiled run-time workflow required of commercial WFMS [Vossen and Weske 1997]. The Java client/server WASA and CORBA-based $WASA_2$ prototypes are designed to dynamically accommodate both anticipated and ad hoc modifications to workflow during computer-based experimentation.

The CRISTAL architecture incorporates a comprehensive workflow system for the production management of a complex scientific instrument [Baker et al. 1997; McClatchey et al. 1997b]. The LabBase project accomplishes laboratory workflow management by tracking the state transitions of objects representing laboratory protocols (methodologies).

Singh and Vouk [1996] view the implementation of complex systems of scientific computations as scientific workflow. They recognize the benefits of applying workflow specification and scheduling to these computational studies or experiments. The authors suggest that introducing workflow support could extend collaboration through the entire data production phase, bringing the benefits of collaborative free-form research workflow to the later stages of opaque and inflexible, fully-automated production workflow.

Research on dynamic and adaptive workflow systems that could aid the development of scientific workflow systems is ongoing, with proposed solutions in one recent workshop based primarily on introducing exception handling to a fixed process model or offering partial specification using, for example, late binding techniques [Bernstein et al. 1999]. Rusinkiewicz and Sheth [1995] discuss a model for transactional workflow systems. In these systems, the techniques of extended and relaxed DBMS transactions are exploited where appropriate, but the authors maintain that these advanced transaction models are too inflexible to serve as a general solution for workflows. In their critique of commercial workflow systems, Alonso et al. [1997a] take the position that, while essential, the "cross-fertilization between advanced transaction models and workflow environments" has yet to play out.

6. RELATED RESEARCH: VERSION MANAGEMENT

Version management, known also as version or revision control, is used for tracking modifications to files or objects (particularly documents, application code components, and database records) over time. Such objects are often subject to concurrent access and modification by a group. The ability to recover a previous version of an object is one motivation for employing version management.

The basic concept in widely used software such as the Concurrent Versions System (CVS) [Cederqvist 1993] is a controlled repository for latest versions, where files or objects are required to be checked out prior to modification. Upon check-in, modified files or objects supercede previously existing versions. For collections of interrelated objects, possibly within multi-user systems, configuration management and change propagation become important [Date 2000].

In CVS, user-entered "log messages" are appended to a version control history file when modified files are committed (checked in) to the repository. The history file, which maintains a log of "what files have changed when, how, and by whom" [Date 2000] is available for browsing. Various other ways to note when changes occur to files are available through user-defined logging in CVS. CVS and similar systems are designed to allow the lossless recovery of any previous version by incrementally rolling back successive edits.

Although they are concerned with software products rather than data products, Conradi and Westfechtel [1998] provide a comprehensive framework of issues to consider when implementing any versioned object base, which they define as the interplay between *product space* and *version space*. The architecture of a system that enables lineage retrieval will depend on the choice of version model and the relationship between the chosen workflow, metadata, and version models. For example, a version model may be implemented on top of a specific workflow model, or vice versa. Or a version model may be an extension or feature of the chosen workflow or metadata model.

Some projects implement version control from a data lineage perspective. In GOOSE, the lineage of graph nodes describing the cooperation between objects is used to manage data object versions for concurrent access [Alonso and El Abbadi 1993]. Spery et al. [1999] propose a lineage metadata model for implementing version control of an existing cadastral database. They investigate using data lineage to manage the propagation of updates from a central land parcel DBMS to distributed user databases. In the model, a land parcel object split into new parcels, for example, is a parent object connected by *filiation links* to its child objects. The links represent changes over time. *Filiation tree* diagrams show the relationship between parcel objects in different time periods. Their primary goal is to preserve the transformations of land parcels over time and enable historical queries.

Other scientific data management systems are also concerned with version control. Versioning and configuration management are an integral part of the CRISTAL project: [McClatchey et al. 1997a; Barry et al. 1998] discuss how to accommodate coexisting versions of part and flow definitions by using workflow *meta-objects* [Object Management Group 2002].

Barkstrom [2002] emphasizes the importance of properly identifying the subtle gradations of configuration in a multilevel version model for batch processing at a NASA Earth science data center. For example, to ensure the validity of scientific research, maintaining the consistency of data product versions through uniform production code configurations is crucial. Minor, consistent variation in computing environments is less important. Barkstrom describes using a pair of arrays to store a prescribed hierarchy of typed nodes and extracting graphs from these arrays for both data collections (provenance graphs) and data production flow (production graphs). This scheme provides the ability to store compact lineage information for both products and flows in one archive.

Table VI.

System	Years	Description/Goals	Refs.
Amazonia	1993–1997	A computational modeling system supporting increased efficiency of scientists in iterative process of modeling.	[Smith et al. 1995; Saran et al. 1996]
CCDB: Computational Chemistry Database	1994	A computational experiment management system providing data management and interoperability for computational science applications.	[Cushing et al. 1994]
CRISTAL: Concurrent Repository and Information System for the Tracking of Assembly Lifecycles	1996–	A large scale distributed scientific workflow management project to track the mechanical processing of crystals destined for the CMS detector.	[Le Goff et al. 1996; Baker et al. 1997; McClatchey et al. 1998; Draskic et al. 1999]
ESP^{2}Net: Earth Science Partners' Private Network	1998–	A collaborative scientific computing environment.	[Kaestle et al. 1999]
Gaea	1992–1994	A scientific DBMS for metadata management supporting geographic information analysis and global change research.	[Hachem et al. 1993]
LabBase	1994	A generic DBMS implementing laboratory information systems, used to manage workflow in large semi-automated laboratory projects.	[Stein et al. 1994]
OPM: Object Protocol Model	1995–1998	An object data model that supports specifying database schemas in terms of objects and laboratory protocols.	[Chen and Markowitz 1995b; Chen and Markowitz et al. 1995a]
Sequoia 2000	1991–1995	A collaboration between computer scientists and environmental researchers to design a next-generation information system for managing data for global change research.	[Stonebraker 1991; Stonebraker 1994]
ViNE: Virtual Notebook Environment	1998–	A platform-independent, web-based interface supporting collaboration and management of computational experiments.	[Skidmore et al. 1998]
WASA: Workflow-based Architecture to support Scientific Applications	1995–1999	A workflow system supporting scientific application environments.	[Medeiros et al. 1995; Vossen and Weske 1997; Vossen and Weske 1999]
ZOO	1989–1998	A desktop experiment management system.	[Ioannidis et al. 1996]

7. RELATED RESEARCH: SUMMARY OF SYSTEMS CONSIDERED

The prototype systems mentioned in Sections 4–6 do not necessarily include the capability of lineage retrieval, but provide a review of workflow and metadata issues. Table VI provides a listing of the major systems discussed in these sections.

8. A METAMODEL FOR LINEAGE RETRIEVAL

A synthesis of the research reviewed in Sections 3–6 provides a framework to clarify the architecture of previous prototypes and ultimately direct the architectural design of new systems with respect to lineage retrieval for the results of data processing.

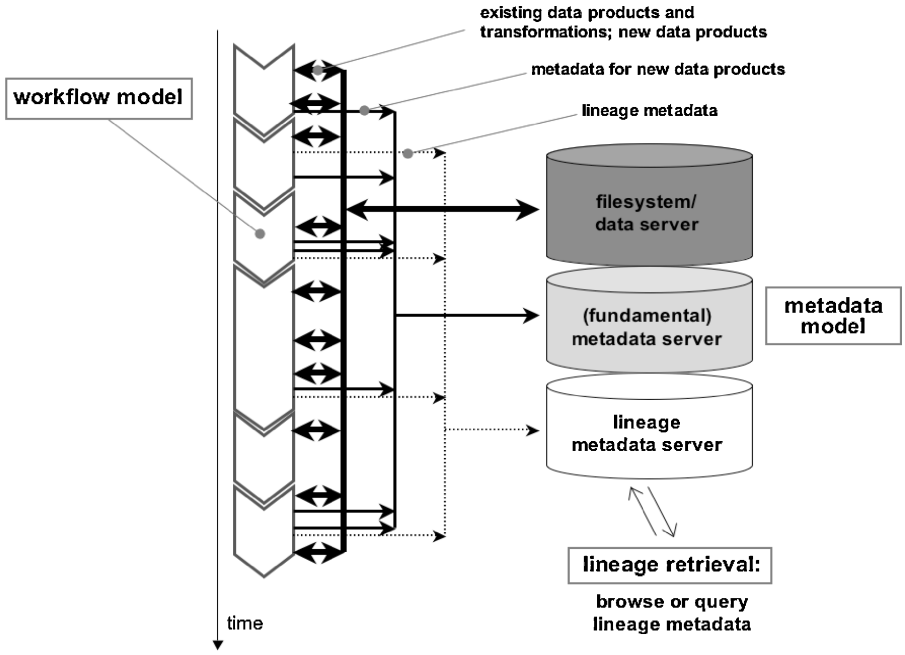

Fig. 4. A metamodel for lineage retrieval.

Figure 4 illustrates the components of a metamodel where the constituents in a workflow model (where model is used in the sense of general abstraction) are delivered to and from a filesystem or other type of repository. If provided, metadata values for the new items generated by the workflow correspond to some metadata model. Lineage retrieval is possible when lineage metadata, describing the relationship between workflow constituents after processing has occurred, is also supplied by the workflow. The data, metadata, and lineage metadata servers to the right of the diagram may or may not be logically or physically distinct.

Figure 5 uses the UML [Booch et al. 1999] package notation to distill the basic framework from Figure 4, conveying the observation that lineage retrieval is dependent on the workflow and metadata models embodied by a particular scientific computing or information system. In short, lineage retrieval requires the capability to assemble a retrospective view of workflow using extant metadata.

For each of the three framework components, a set of considerations is presented to guide the description of four recent prototype systems. The four systems selected—ESSW, Geo-Opera, MyGrid, and Chimera—all share an explicit goal of tracking and retrieving lineage for new scientific data products.

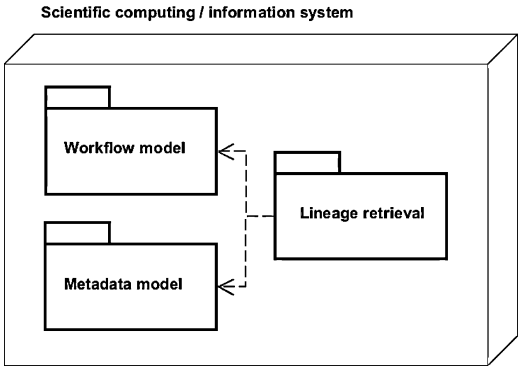

Fig. 5. A metamodel for lineage retrieval: UML package notation.

8.1. Workflow Model

Here the term model is used in the broad sense of abstraction and description; that is, this section investigates how data

processing workflow is abstracted and described (the italicized terms in each section) in the prototype systems considered.

The following questions serve as general guidelines for the different system descriptions included in this section:

—How is workflow described? What data model (e.g., graph, tree, relation, list) is used for workflow? What items does workflow comprise?

—What identifiers are used for workflow constituents? How are identifiers supplied for new items created by the workflow?

—How is workflow invoked?

—How are the relationships between workflow constituents stored by the system before workflow invocation? That is, how is the workflow data model implemented by the system?

8.1.1. ESSW. In ESSW, a *model* is a DAG of inputs/outputs and experiment steps. An *experiment* is an instance of this model that consists of input/output and experiment step *science objects* residing in a filesystem. Before an experiment can be run or invoked, class definitions or templates (XML DTDs) for all science objects in the experiment must be submitted to the ESSW Lab Notebook. When an experiment involving existing science objects is invoked by running the required (Perl) wrapper scripts, the system creates records in the Lab Notebook relational DBMS (RDBMS), using system-generated object identifiers (OIDs), in order to track existing science object instances. Records for newly created science objects are also generated by the system.

Relationships between workflow constituents are not known to the system prior to running an experiment. Parent/child relationships between science objects are explicitly described in the wrapper scripts and are only communicated to ESSW after an experiment has been run.

8.1.2. Geo-Opera. A *geo-process* in Geo-Opera is modeled as a set of connected *tasks* (activities, blocks, or submodels). Task connectors correspond internally to *guards*, or rules for task execution. Input parameters and return values for a task—that is, the relationships between workflow constituents—are specified when the task is registered with the system. The *activity* task type corresponds to an external program that uses or creates (data) *objects*.

As in ESSW, internal objects with system-generated OIDs are created when external data is registered with Geo-Opera, although in this case the system predefines attributes for objects that are not modifiable by the user. Workflow is invoked when a program written in an application-specific Geo-Opera language is executed.

In Geo-Opera, internal objects have a Sources attribute: a list of immediate ancestor OIDs which is updated by the system during processing. There is also an Algorithm attribute that keeps a record of the task used to create the object, and a Usage attribute: a list of tasks that use the object as input.

8.1.3. myGrid. A *workflow* for myGrid, which uses the IT Innovation workflow enactment engine [IT Innovation 2002], consists of services and their *inputs* and *outputs*. Workflow enactment XML scripts, which encode the relationships between the workflow input and output constituents, are stored in the myGrid Information Repository (mIR) along with the inputs and outputs themselves with a system-generated Life Sciences Identifier [Object Management Group 2004]. A workflow is invoked when a workflow script is executed.

8.1.4. Chimera. The Virtual Data Schema employed by the Chimera system draws a distinction between an abstract DAG defined by the interplay between general *transformations* and argument *types* and a concrete DAG consisting of an invocation of transformation instances (*derivations*) and replicas of type instances (*datasets*). Similar to the MyGrid system, relationships between workflow constituents are encoded in the Virtual

Data Language (VDL) wrapper scripts that define concrete DAGs. These VDL scripts are submitted to the Virtual Data Catalog (VDC). The author of a VDL script is responsible for maintaining unique names or identifiers and any versioning information for workflow constituents. Workflow is invoked when a so-called DAX (abstract DAG in XML) file—a logical workflow plan generated by the Chimera abstract planner from the VDC and a catalog of transformations—is submitted to the Grid.

8.2. Metadata Model

The following questions serve as general guidelines for the different system descriptions concerning the abstraction and description of fundamental and lineage-related metadata for data processing presented in this section:

—What fundamental metadata, if any, is supplied for workflow constituents? How is this fundamental metadata defined? How are values added to this metadata? How is fundamental metadata generated for new items created by a workflow invocation?

—What type of container is used for fundamental metadata? How/where is this metadata stored?

—What lineage metadata, if any, is created during a workflow invocation? How is this lineage metadata created?

—What type of container is used for lineage metadata? How/where is this metadata stored?

8.2.1. ESSW. In ESSW, the idea of metadata and workflow are intertwined because, unlike other prototypes, recording the invocation of a workflow depends on metadata values sent by wrapper scripts. The XML DTDs required to register classes of science objects in ESSW supply user-defined metadata for each class of object. When an experiment is run, the wrapper scripts send metadata values to the Lab Notebook which creates XML for an object based on the DTD and the corresponding values sent. The XML metadata created for a science object by the system is then partially parsed, with some elements stored as separate attributes in a new record in the RDBMS table for the science object, and with the entire block of XML stored as a binary large objecct (BLOB) in that record.

The wrapper scripts for an experiment are responsible for sending experiment step input and output information to the Lab Notebook. The system populates a binary (i.e., two attribute) relation in the Lab Notebook RDBMS table with the science object identifiers to represent parent/child relationships received from the scripts.

8.2.2. Geo-Opera. Geo-Opera activity tasks and objects represent programs and data external to the system. Standard interfaces with predefined attributes are defined for both tasks and objects, but fundamental metadata is not accommodated within the system.

The system updates lineage-related attributes for objects when models are run with the execution of scripts. Data object attributes reside in the system object space (DBMS).

8.2.3. myGrid. mIR objects have provenance attributes and hold metadata, some of which is created manually beforehand, and some of which is generated by the workflow invocation. User-supplied "ws-info" (workflow service information) documents describe workflow constituents in terms of predefined ontologies.

The workflow enactment script creates a provenance log for each enactment, and metadata, such as service execution start and stop times, is added to the provenance log. These logs and other items are stored in the mIR. The workflow constituents described in provenance logs are either manually or automatically annotated with additional semantic information drawn from the ws-info documents after processing has occurred.

8.2.4. Chimera. Fundamental metadata is not directly included in the current release (1.2) of the Chimera system. Lineage

metadata is stored implicitly in VDL script and is interpreted when the DAX logical workflow plan is created. The logical workflow plan defines a concrete DAG which is submitted to the Grid.

8.3. Lineage Retrieval

The following questions serve as general guidelines for descriptions of the lineage retrieval methods used by the prototype systems considered:

—How can one overview or browse the lineage of an item in the system?
—What methods are used for lineage retrieval?
—What type of lineage-related queries can be asked of the system? How are these queries submitted?
—How are the results of lineage queries displayed?

8.3.1. ESSW. Any Web browser can access an application created to display the lineage of a science object recorded in ESSW. A user browses to, or enters the ID of, an item and specifies the number of levels of forward or backward lineage to retrieve. The application retrieves the lineage using recursive SQL queries on the lineage parent/child RDBMS table. The lineage is displayed as a (GraphViz [AT&T 2001]) graph of science objects. Clicking any science object in the graph displays the fundamental metadata for that object.

8.3.2. Geo-Opera. [Alonso and Hagen 1997b] describes an example of a Geo-Opera user creating an error calculation algorithm for a process through lineage query facilities provided by the system. The system itself also uses these facilities to maintain internal consistency.

8.3.3. myGrid. The MyGrid project has demonstrated the ability to bring items from the mIR into the COHSE conceptual open hypermedia system to allow a user to browse provenance links created manually or automatically by the system. Ontology terms are used to associate concepts with entries in the provenance logs.

8.3.4. Chimera. When a concrete DAG is submitted to the Grid via a launcher program, the launcher creates a record of the DAG invocation in the currently experimental Provenance Tracking Catalog (PTC). This store would provide information about what actually occurred during invocation, including any remote job failures.

The Chimera VDL includes commands for querying a VDC about the existence of derived data [Foster et al. 2002], but this is only the plan for a workflow invocation, not true lineage as with the PTC. The Chimera project envisions distributed VDCs capable of inter-catalog references and provenance chains that span across servers with provenance hyperlinks [Foster et al. 2003].

8.4. Prototype Systems: Discussion

As shown in the framework of Figure 4, the four prototypes share common components. These include: invoking workflows through scripts; using data and metadata repositories to track workflow constituents and their relationships; and retrieving lineage by relying on a trail of metadata to reassemble the record of workflow.

The descriptions in the previous sections also reflect differences in the approaches to data processing that the systems were designed for. Geo-Opera is concerned with addressing some of the deficiencies of commercial enterprise WFMS, myGrid and Chimera are designed to construct workflows of Grid services, and ESSW is concerned with the script-based processing favored by portions of the environmental and Earth Science communities. The following discussion includes implications for the design of new systems for the same scripted application domain as ESSW.

8.4.1. Workflow. Although freedom from overstructured programming is one of the hallmarks of scripting environments, the perspective of data processing as a graph of uniquely identifiable workflow constituents is critical for improving the

documentation of scientific research computing. Experience [Frew and Dozier 1997; Frew and Bose 2001] informs us, however, that researchers may resist systems or methods that encroach upon existing scripting methodology or that pose too great of an administrative burden on a research group. New systems designed to retrieve the lineage of data products need to provide a general workflow model that serves to organize the creation and management of scripts without being too restrictive.

The four sets of workflow terminology for the four systems described demonstrates the lack of a definitive view of data processing workflow in the sciences, a situation which is likely to persist. Systems that provide access to lineage will benefit by incorporating the ability to accommodate varied metadata standards and the new methods for ontology construction that will help others to determine the composition of workflows.

8.4.2. Metadata. ESSW requires users to create XML DTDs in order to provide metadata attributes for each object they require for their processing. This procedure was esoteric enough to create difficulties in prototype case studies and required the intercession of staff programmers. Newer systems with XML components may need to develop friendlier tools similar to Morpho, an interface to assist ecologists create XML metadata for their data sets using standard, domain-specific terms developed as part of the Knowledge Network for Biocomplexity (KNB) project [Berkley et al. 2001].

Interestingly, Geo-Opera is not designed to provide fundamental metadata for workflow constituents. While the Chimera project envisions the need for VDCs to integrate with metadata catalogs and other information resources (see Foster et al. [2003], Section 4.1), the current release of the system does not allow for this [Grid Physics Network (GriPhyN) project 2003].

Woodruff and Stonebraker [1997] and Marathe [2001] are notable for exploring alternatives to the use of fundamental metadata for examples of retrieving fine-grained lineage (specific elements from source arrays) for the results of array processing. However, this capability is only available for processing performed within these particular systems. Coarse grain fundamental metadata for transformations and interim data products will still be useful for providing standard documentation for more complex, multiform data processing.

8.4.3. Lineage Retrieval. The demonstration of provenance browsing in the myGrid project with COHSE, together with the similar concept of the pedigree browser put forward by the CMCS project [Pancerella et al. 2003], suggests the utility of a web of references and verifiable annotations to supplement the line of direct ancestor and descendent workflow constituents that forms the basis of lineage.

The multiple benefits of the ability to retrieve lineage have been cited by many researchers (Tables III and IV), but the small number of operational examples created to date for browsing and querying the lineage of data processing results implies that this is still an open research area. Exploring the interfaces and visualization techniques used by increasingly popular genealogy software and investigating the adaptation of those techniques to represent the lineage of scientific data products and transformations may prove useful.

9. CONCLUSION

Discussions at two data provenance workshops [Buneman and Foster 2002a, 2003] show that researchers and data managers in a variety of scientific disciplines are recognizing the imperative of providing lineage metadata for their custom data products to research partners or other potential data consumers. However, these meetings also reveal that operational systems to achieve this are not yet widespread and that the results of previous research in this area are still being assimilated.

To organize our review, we recognized several broad categories for data

processing where the processing flow is controlled by command line entries, scripts or programs, DBMS queries, WFMS directives, or Web- or Grid services. It is also realistic to consider scientific data processing workflows that involve various combinations of these categories. Because of this, interest in systems to help assemble scientific workflows continues. Lineage retrieval for such complex workflows will clearly present challenges.

The Internet has heightened expectations for data consumers' immediate gratification, including the ability to:

—access, select, and download programs and data products at will, and

—track the status or shipping history of packages or other items at will.

In addition, low-cost commercial genealogy software now provides the ability to:

—easily create complex lineage charts, reports, and queries (for family trees).

At its best, lineage retrieval for scientific data processing will provide a mixture of these three concepts. Systems including lineage retrieval will ideally allow a potential data consumer to access and select a custom scientific program or data product and browse or query the lineage or processing history using comprehensive charts or diagrams before downloading. The work discussed in this survey serves as a nascent effort to provide these capabilities to scientists and to potential consumers of their data processing results.

ACKNOWLEDGMENTS

We would like to acknowledge the constructive comments of two anonymous reviewers, as well as helpful discussions with Bruce Barkstrom and participants of the Workshop on Data Derivation and Provenance, October 17-18, 2002, Chicago, IL, and the Workshop on Data Provenance and Annotation, December 1–3, 2003, Edinburgh, Scotland, both organized by Peter Buneman and Ian Foster.

REFERENCES

ALONSO, G. 1994. Managing advanced databases: Concurrency, recovery, and cooperation in scientific applications. Ph.D. Dissertation, Computer Science Department, University of California at Santa Barbara, Santa Barbara, CA.

ALONSO, G., AGRAWAL, D., EL ABBADI, A., AND MOHAN, C. 1997a. Functionality and limitations of current workflow management systems. Computer Science Department, University of California at Santa Barbara, Santa Barbara, CA. Available at: http://www.inf.ethz.ch/personal/alonso/PAPERS/IEEE-Expert.ps.Z.

ALONSO, G., AND EL ABBADI, A. 1993. GOOSE: Geographic object oriented support environment. In *Proceedings of the ACM Workshop on Advances in Geographic Information Systems*. Arlington, VA. 38–49.

ALONSO, G., AND HAGEN, C. 1997b. Geo-Opera: Workflow concepts for spatial processes. In *Proceedings of the 5th International Symposium on Spatial Databases (SSD '97)*. Berlin, Germany. 238–258.

ALONSO, G., HAGEN, C., SCHEK, H.-J., AND TRESCH, M. 1998. Towards a platform for distributed application development. In *Workflow Management Systems and Interoperability*. A. Dogac, L. Kalinichenko, M. T. Ozsu and A. Sheth, Eds. NATO ASI Series, Vol. 164. Springer, Berlin. 195–221.

AOYAMA, M., WEERAWARANA, S., MARUYAMA, H., SZYPERSKI, C., SULLIVAN, K., AND LEA, D. 2002. Web services engineering: promises and challenges. In *IEEE Proceedings of the 24th International Conference on Software Engineering (ICSE '02)*. Orlando, FL. 647–648.

AT&T. 2001. Graphviz graph visualization software. AT&T Labs—Research. Available at: http://www.research.att.com/sw/tools/graphviz/.

BAKER, N., MCCLATCHEY, R., AND LE GOFF, J.-M. 1997. Scientific workflow management in a distributed production environment. In *IEEE Proceedings of the 1st International Enterprise Distributed Object Computing Workshop*. 291–299.

BARKSTROM, B. R. 1998. Digital archive issues from the perspective of an Earth Science data producer. Position Paper: ISO Archiving Workshop Series: Digital Archive Directions (DADs) Workshop (June). College Park, MD. Available at: http://ssdoo.gsfc.nasa.gov/nost/isoas/dads/.

BARKSTROM, B. R. 2002. Data product configuration management and versioning in large-scale production of satellite scientific data production. Position paper: Workshop on Data Derivation and Provenance (Oct.). Chicago, IL.

BARRY, A., BAKER, N., LE GOFF, J.-M., MCCLATCHEY, R., AND VIALLE, J.-P. 1998. Meta-data based design of workflow systems. Workshop paper: Metadata and Dynamic Object-Model Pattern Mining Workshop (at *OOPSLA '98*) (Oct.). Vancouver, Canada. Available at: http://www-poleia.lip6.fr/~razavi/aom/papers/oopsla98/mcclatchey.pdf.

BECKER, R. A., AND CHAMBERS, J. M. 1988. Auditing of data analyses. *SIAM J. Sci. Stat. Comput. 9*, 4, 747–760.

BERKLEY, C., JONES, M., BOJILOVA, J., AND HIGGINS, D. 2001. Metacat: A schema-independent XML database system. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management (SSDBM '01)* (July), Fairfax, VA, L. Kerschberg and M. Kafatos, Eds. IEEE Computer Society. 171–179.

BERNSTEIN, A., DELLAROCAS, C., AND KLEIN, M. 1999. Towards adaptive workflow systems. *SIGMOD Record 28*, 3, 7–8.

BOOCH, G., RUMBAUGH, J., AND JACOBSON, I. 1999. *The Unified Modeling Language User Guide*. Addison-Wesley.

BROWN, P., AND STONEBRAKER, M. 1995. Big Sur: A system for the management of Earth science data. In *Proceedings of the 21st International Conference of Very Large Data Bases (VLDB '95)*. Zurich, Switzerland. 720–728.

BUNEMAN, P., AND FOSTER, I. 2002a. Workshop on Data Derivation and Provenance. (Oct). Chicago, IL. Available at: http://www-fp.mcs.anl.gov/~foster/provenance/.

BUNEMAN, P., AND FOSTER, I. 2003. Workshop on Data Provenance and Annotation (Dec.). Edinburgh, Scotland. Available at: http://www.nesc.ac.uk/esi/events/304/.

BUNEMAN, P., KHANNA, S., AND TAN, W. C. 2000a. Data provenance: Some basic issues. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science (FSTTCS '00)*. New Delhi, India. Springer, 87–93.

BUNEMAN, P., KHANNA, S., AND TAN, W. C. 2001. Why and where: A characterization of data provenance. In *Proceedings of the International Conference on Database Theory (ICDT '01)* (Jan.). London, UK. 316–330.

BUNEMAN, P., KHANNA, S., AND TAN, W. C. 2002b. Computing provenance and annotations for views. Workshop Paper: Workshop on Data Derivation and Provenance (Oct.). Chicago IL. Available at: http://people.cs.uchicago.edu/~yongzh/position_papers.html.

BUNEMAN, P., MAIER, D., AND WIDOM, J. 2000b. Where was your data yesterday, and where will it go tomorrow? Data Annotation and Provenance for Scientific Applications. Position paper for NSF Workshop on Information and Data Management (IDM '00): Research Agenda into the Future (March), Chicago IL.

CEDERQVIST, P. 1993. Version management with CVS, Signum Support AB (Dec.). Available at: https://www.cvshome.org/docs/manual/.

CHAKRAVARTHY, S., KRISHNAPRASAD, V., TAMIZUDDIN, Z., AND LAMBAY, F. 1993. A federated multi-media DBMS for medical research: Architecture and functionality. Technical Report UF-CIS-TR-93-006, Department of Computer and Information Sciences, University of Florida, Gainesville, FL.

CHEN, I. A., AND MARKOWITZ, V. M. 1995a. Modeling scientific experiments with an object data model. In *Proceedings of the 11th International Conference on Data Engineering (ICDE '95)*. 391–400.

CHEN, I. A., AND MARKOWITZ, V. M. 1995b. An overview of the Object Protocol Model (OPM) and the OPM data management tools. *Inform. Syst. 20*, 5, 393–418.

CHEN, L., SHADBOLT, N. R., GOBLE, C., TAO, F., COX, S. J., PULESTON, C., AND SMART, P. 2003. Towards a knowledge-based approach to semantic service composition. Lecture Notes in Computer Science. *2870*, 319–334.

CICHOCKI, A., HELAL, A., RUSINKIEWCZ, M., AND WOELK, D. 1998. *Workflow and Process Automation*. Kluwer Academic Publishers, London, UK.

CLARKE, D. G., AND CLARK, D. M. 1995. Lineage. In *Elements of Spatial Data Quality*, S. C. Guptill and J. L. Morrison, Eds., Elsevier Science, Oxford. 13–30.

CONRADI, R., AND WESTFECHTEL, B. 1998. Version models for software configuration management. *ACM Comput. Sur. 30*, 2, 232–282.

CUI, Y., AND WIDOM, J. 2003. Lineage tracing for general data warehouse transformations. *The VLDB J. 12*, 1, 41–58.

CUI, Y., WIDOM, J., AND WIENER, J. L. 1997. Tracing the lineage of view data in a warehousing environment. Technical Report, Stanford University Database Group (Nov.). Stanford, CA. Available at: http://www-db.stanford.edu/pub/papers/lineage-full.ps.

CUI, Y., WIDOM, J., AND WIENER, J. L. 2000. Tracing the lineage of view data in a data warehousing environment. *ACM Trans. Datab. Syst. 25*, 2, 179–227.

CUSHING, J. B., MAIER, D., RAO, M., ABEL, D., FELLER, D., AND DEVANEY, D. M. 1994. Computational proxies: Modeling scientific applications in object databases. In *Proceedings of the 7th International Working Conference on Scientific and Statistical Database Management (SSDBM '94)*. 196–206.

DATE, C. J. 2000. *Introduction to Database Systems*. Addison-Wesley.

DRASKIC, J., LE GOFF, J.-M., WILLERS, I., ESTRELLA, F., KOVACS, Z., MCCLATCHEY, R., AND ZSENEI, M. 1999. Using a meta-model as the basis for enterprise-wide data navigation. In *Proceedings of the 3rd IEEE Metadata Conference (MD'99)* (April). Bethesda, MO.

EAGAN, P. D., AND VENTURA, S. J. 1993. Enhancing value of environmental data: data lineage reporting. *J. Environ. Eng. 119*, 1, 5–16.

ELMAGARMID, A., AND DU, W. 1997. Workflow management: State of the art versus state of the products. In *Workflow Management Systems and Interoperability*, A. Dogac, L. Kalinichenko, M. T. Ozsu and A. Sheth, Eds. NATO ASI Series, Vol. 164, Springer, Berlin. 1–17.

ESRI. 1982. ARC/INFO geographic information system (GIS), ESRI, Redlands, CA. Available at: www.esri.com.

FEDERAL GEOGRAPHIC DATA COMMITTEE. 1998. Content standard for digital geospatial metadata

FGDC-STD-001-1998 (revised June), Federal Geographic Data Committee, Washington, DC. Available at: http://www.fgdc.gov/metadata/csdgm/.

FELDMAN, S. I. 1978. Make—A program for maintaining computer programs. In *UNIX Programmer's Manual*, Vol. 2 (Bell Laboratories). Holt, Rinehart and Winston, New York. 291–300.

FOSTER, I., AND KESSELMANN, C., Eds. 1999. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

FOSTER, I., VOCKLER, J., WILDE, M., AND ZHAO, Y. 2002. Chimera: A virtual data system for representing, querying, and automating data derivation. In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM '02)* (July). Edinburgh, Scotland, J. Kennedy, Ed. IEEE Computer Society. 37–46.

FOSTER, I., VOCKLER, J., WILDE, M., AND ZHAO, Y. 2003. The virtual data grid: A new model and architecture for data-intensive collaboration. In *Proceedings of the 1st Biennial Conference on Innovative Data System Research (CIDR '03)* [Online proceedings] (Jan.). Pacific Grove, CA.

FRENCH, J. C. 1995. What is metadata? In *Proceedings of the SDM–92 Workshop: The Role of Metadata in Managing Large Environmental Science Datasets,* Richland, WA, R. B. Melton, D. M. DeVaney and J. C. French, Eds. Pacific Northwest Laboratory. 3–8.

FREW, J., AND BOSE, R. 2001. Earth system science workbench: A data management infrastructure for earth science products. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management (SSDBM '01)* (July). Fairfax, VA. L. Kerschberg and M. Kafatos, Eds. IEEE Computer Society. 180–189.

FREW, J., AND DOZIER, J. 1997. Data management for earth system science. *SIGMOD Record 26*, 1, 27–31.

GEIST, A., AND NACHTIGAL, N. 2003. ORNL Electronic Notebook Project. Oak Ridge National Laboratory. Available at: http://www.csm.ornl.gov/~geist/java/applets/enote/.

GEOGRAPHIC DESIGNS. 1993. Geolineus Version 3.0 User Manual. Santa Barbara, CA.

GEORGAKOPOULOS, D., HORNICK, M., AND SHETH, A. 1995. An overview of workflow management: from process modeling to workflow automation infrastructure. *Distrib. Paral. Datab. 3*, 2, 119–153.

GOLAND, Y., WHITEHEAD, E., FAIZI, A., CARTER, S., AND JENSEN, D. 1999. HTTP Extensions for distributed authoring–WEBDAV: RFC 2518. Network Working Group. Available at: http://asg.web.cmu.edu/rfc/rfc2518.html.

GREENWOOD, M., GOBLE, C., STEVENS, R., ZHAO, J., ADDIS, M., MARVIN, D., MOREAU, L., AND OINN, T. 2003. Provenance of e-science experiments—experience from bioinformatics. In *Proceedings of the UK e-Science All Hands Meeting.* Nottingham, UK. 223–226.

GRID PHYSICS NETWORK (GRIPHYN) PROJECT. 2003. Chimera Virtual Data System Version 1.2 User Guide, Grid Physics Network (GriPhyN) project (Dec.). Available at: http://www.griphyn.org/chimera/release.html.

HACHEM, N. I., QUI, K., GENNERT, M., AND WARD, M. 1993. Managing derived data in the Gaea scientific DBMS. In *Proceedings of the 19th International Conference on Very Large Databases (VLDB '93)* (Aug.). Dublin, Ireland. 1–12.

INSIGHTFUL CORPORATION. 2003. S-PLUS statistical analysis, graphics and programming application, Insightful Corporation, Seattle, WA. Available at: http://www.insightful.com/.

IOANNIDIS, Y., LIVNY, M., GUPTA, S., AND PONNEKANTI, N. 1996. ZOO: A desktop experiment management environment. In *Proceedings of the 22nd International Conference on Very Large Databases (VLDB '96).* Bombay, India. 274–285.

IOANNIDIS, Y., LIVNY, M., HABER, E., MILLER, R., TSATALOS, O., AND WIENER, J. 1993. Desktop experiment management. *IEEE Data Eng. Bull. 16*, 1, 19–23.

IT INNOVATION. 2002. IT innovation workflow enactment engine. IT Innovation Centre. Available at: http://www.it-innovation.soton.ac.uk/mygrid/workflow/.

KAESTLE, G., EDDIE C. SHEK, AND DAO, S. K. 1999. Sharing experiences from scientific experiments. In *Proceedings of the 11th International Conference on Scientific and Statistical Database Management (SSDBM '99)* (July). Cleveland, OH. IEEE Computer Society, 168–177.

KAVANTZAS, N., BURDETT, D., AND RITZINGER, G. 2004. Web Services Choreography Description Language Version 1.0. W3C Working Draft, IBM developerWorks (April). Available at: http://www.w3.org/TR/ws-cdl-10/.

LANTER, D. P. 1988. A neural network for GIS command language translation. Unpublished research paper. University of South Carolina, Columbia, SC.

LANTER, D. P. 1989a. Techniques and methods of spatial data-base lineage tracing. Ph.D. Dissertation, University of South Carolina, Columbia, SC.

LANTER, D. P. 1989b. Trimming Large spatial databases with lineage analysis. In *Proceedings of the 10th Annual ESRI Users Conference*. Palm Springs, CA.

LANTER, D. P. 1990. Lineage in GIS: The problem and a solution. Technical Report 90-6, National Center for Geographic Information and Analysis (NCGIA), University of California at Santa Barbara, Santa Barbara, CA.

LANTER, D. P. 1991. Design of a lineage-based meta-data base for GIS. *Cart. Geograph. Info. Syst. 18*, 4, 255–261.

LANTER, D. P. 1993. A Lineage meta-database approach toward spatial analytic database optimization. *Cart. Geograph. Info. Syst. 20*, 2, 112–121.

LANTER, D. P. 1994. Comparison of spatial analytic applications of GIS. In *Environmental Information Management and Analysis: Ecosystem to Global Scales*, W. K. Michener, J. W. Brunt and S. G. Stafford, Eds. Taylor & Francis, Bristol, PA. 413–425.

LANTER, D. P., AND VEREGIN, H. 1990. A lineage meta-database program for propagating error in geographic information systems. In *Proceedings of the GIS/LIS Conference* (Nov.). 144–153.

LE GOFF, J.-M., VIALLE, J.-P., BAZAN, A., LE FLOUR, T., LIEUNARD, S., ROUSSET, D., MCCLATCHEY, R., BAKER, N., KOVACS, Z., HEATH, H., LEONARDI, E., BARONE, G., AND ORGANTINI, G. 1996. C. R. I. S. T. A. L./ Concurrent repository & information system for tracking assembly and production lifecycles—A data capture and production management tool for the assembly and construction of the CMS ECAL detector. CERN CMS Note 1996/003, CERN, 1996, Geneva, Switzerland. Available at: http://cmsdoc.cern.ch/documents/96/note96_003.pdf.

LEE, J., GRUNINGER, M., JIN, Y., MALONE, T., TATE, A., AND YOST, G. 1998. PIF The process interchange format. In *Handbook on Architectures of Information Systems*. P. Bernus, G. Schmidt and K. Mertins, Eds. Springer, Berlin. 167–189.

MANOLA, F., AND MILLER, E. 2004. RDF Primer W3C Recommendation. World Wide Web Consortium (W3C). Available at: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

MARATHE, A. P. 2001. Tracing lineage of array data. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management (SSDBM '01)* (July). Fairfax, VA. L. Kerschberg and M. Kafatos, Eds. IEEE Computer Society. 69–78.

MATHWORKS. 2003. MATLAB programming and visualization application. The Mathworks, Inc., Natick, MA. Available at: http://www.mathworks.com/.

MCCLATCHEY, R., BAKER, N., HARRIS, W., LE GOFF, J.-M., KOVACS, Z., ESTRELLA, F., BAZAN, A., AND LE FLOUR, T. 1997a. Version management in a distributed workflow application. In *IEEE Proceedings of the 8th International Workshop on Database and Expert Systems Applications (DEXA '97)*. 10–15.

MCCLATCHEY, R., ESTRELLA, F., LE GOFF, J.-M., KOVACS, Z., AND BAKER, N. 1997b. Object databases in a distributed scientific workflow application. In *Proceedings of the 3rd Basque International Workshop on Information Technology (BIWIT '97)*. 11–21.

MCCLATCHEY, R., KOVACS, Z., ESTRELLA, F., LE GOFF, J.-M., CHEVENIER, G., BAKER, N., LIEUNARD, S., MURRAY, S., LE FLOUR, T., AND BAZAN, A. 1998. The integration of product data and workflow management systems in a large scale engineering database application. In *IEEE Proceedings of the International Database Engineering and Applications Symposium (IDEAS '98)*. 296–302.

MEDEIROS, C. B., VOSSEN, G., AND WESKE, M. 1995. WASA: A workflow-based architecture to support scientific database applications. In *Proceedings of the 6th International Workshop on Database and Expert Systems Applications (DEXA '95)*. 574–583.

MERRIAM-WEBSTER INC. 2001. *Merriam-Webster Collegiate Dictionary*, Springfield, MA.

MOHAN, C. 1997. Recent Trends in workflow management products, standards and research. In *Workflow Management Systems and Interoperability*. A. Dogac, L. Kalinichenko, M. T. Ozsu and A. Sheth, Eds. NATO ASI Series Vol. 164, Springer. 396–409.

MYERS, J., PANCERELLA, C., LANSING, C., SCHUCHARDT, K., AND DIDIER, B. 2003a. Multi-scale science: Supporting emerging practice with semantically derived provenance. In *Proceedings of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data* [Online proceedings] (Oct.). Sanibel Island, FL. 2003.

MYERS, J. D., CHAPPELL, A. R., ELDER, M., GEIST, A., AND SCHWIDDER, J. 2003b. Re-integrating the research record. *Comput. Sci. Eng. 5*, 3, 44–50.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). 1986. Report of the EOS Data Panel, Vol. IIa: Earth Observing System Data and Information System. Technical Memorandum 87777, National Aeronautics and Space Administration (NASA), Washington, DC.

NATIONAL RESEARCH COUNCIL. 1999. *Global Environmental Change: Research Pathways for the Next Decade*. National Academy Press, Washington, DC.

OBJECT MANAGEMENT GROUP. 2002. Meta-Object Facility (MOF) Specification, Version 1.4. Object Management Group (OMG). Available at: http://www.omg.org/cgi-bin/doc?formal/2002-04-03.

OBJECT MANAGEMENT GROUP. 2004. dtc/04-05-01 (Life Sciences Identifiers final adopted specification). Object Management Group, Inc. Available at: http://www.omg.org/docs/dtc/04-05-01.pdf.

OUSTERHOUT, J. 1994. *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, MA.

PANCERELLA, C., MYERS, J., ALLISON, T. C., AND AMIN, K. 2003. Metadata in the collaboratory for multi-scale chemical science. In *Proceedings of the Dublin Core Conference (DC-'03)* [Online proceedings] (Sept.-Oct.). Seattle, WA.

PRATT, J. M. 1995. Data modeling of scientific experimentation. In *Proceedings of the 1995 ACM Symposium on Applied Comput.*, 86–90.

RESEARCH SYSTEMS INC. 2003. Interactive Data Language (IDL) computing environment for interactive analysis and visualization of data. Research Systems, Inc. Available at: http://www.rsinc.com/.

ROUSH, G. E. 1989. Documenting one's work. *IEEE Potentials 8*, 2, 24–26.

RUSINKIEWICZ, M., AND SHETH, A. 1995. Specification and execution of transactional workflows. In *Modern Database Systems: The Object Model, Interoperability, and Beyond*. W. Kim, Ed. ACM Press, New York. 592–620.

SARAN, A., AGRAWAL, D., EL ABBADI, A., SMITH, T. R., AND SU, J. 1996. Scientific modeling using distributed resources. In *Proceedings of the 4th ACM Workshop on Advances on Advances in Geographic Information Systems,* Rockville, MD. ACM Press. 68–75.

SCHAEL, T. 1998. *Workflow Management Systems for Process Organizations*. Springer, Berlin.

SINGH, M., AND VOUK, M. A. 1996. Scientific workflows: Scientific computing meets transactional workflow. In *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions* [Online Proceedings] (May). Athens, GA.

SKIDMORE, J. L., SOTTILE, M. J., CUNY, J. E., AND MALONEY, A. D. 1998. A prototype notebook-based environment for computational tools. In *IEEE Proceedings of the Supercomputing '98 (SC '98) Conference* (Nov.). Orlando, FL. 7–13.

SMITH, T. R., SU, J., AGRAWAL, D., AND EL ABBADI, A. 1993. Database and modeling systems for the earth sciences. *IEEE Bull. Tech. Comm. Data Eng. 16*, 1, 33–37.

SMITH, T. R., SU, J., EL ABBADI, A., AGRAWAL, D., ALONSO, G., AND SARAN, A. 1995. Computational modeling systems. *Info. Syst. 20*, 2, 127–153.

SPERY, L., CLARAMUNT, C., AND LIBOUREL, T. 1999. A lineage metadata model for the temporal management of a cadastre application. In *Proceedings of the 10th International Workshop on Database and Expert Systems Applications (DEXA '99)* (Sept.). Florence, Italy, A. Cammelli, A. Tjoa and R. R. Wagner, Eds. IEEE Computer Society, 466–474.

STEIN, L., ROZEN, S., AND GOODMAN, N. 1994. Managing laboratory flow with LabBase. In *Proceedings of the Conference on Computers in Medicine (CompMed'94)*.

STONEBRAKER, M. 1991. An overview of the Sequoia 2000 project. Sequoia Technical Report S2K-94-58. Berkeley, CA. Available at: http://epoch.cs.berkeley.edu:8000/sequoia/tech-reports/s2k-94-58/.

STONEBRAKER, M. 1994. Sequoia 2000-a reflection on the first three years. Sequoia Technical Report S2K-94-58. Berkeley, CA. Available at: http://epoch.cs.berkeley.edu:8000/sequoia/tech-reports/s2k-93-23/.

STONEBRAKER, M., CHEN, J., NATHAN, N., PAXSON, C., AND WU, J. 1993. Tioga: Providing data management support for scientific visualization applications. In *Proceedings of the 19th International Conference on Very Large Databases (VLDB '93)*. Dublin, Ireland. 25–38.

THATTE, S. 2003. Business Process Execution Language for Web Services Version 1.1. Specification, IBM developerWorks (May). Available at: http://www-106.ibm.com/developerworks/library/ws-bpel/.

U.S. GEOLOGICAL SURVEY. 1992. Spatial Data Transfer Standard (SDTS) NCITS 320-1998, American National Standards Institute (ANSI) (June). Reston, VA. Available at: http://mcmcweb.er.usgs.gov/sdts/SDTS_standard_nov97/part1b12.html.

U.S. GEOLOGICAL SURVEY. 1995. Modern Average Global Sea-Surface Temperature: Metadata. U.S. Geological Survey. Available at: http://geo-nsdi.er.usgs.gov/metadata/digital-data/10/metadata.html#2.

UC BERKELEY. 1994. POSTGRES database management system (DBMS), Universtity of California Berkeley, Berkeley, CA. Available at: http://db.cs.berkeley.edu/postgres.html.

VAHDAT, A., AND ANDERSON, T. 1998. Transparent result caching. In *Proceedings of the USENIX Annual Technical Conference* [Online proceedings] (June). New Orleans, LA. 1998.

VOSSEN, G., AND WESKE, M. 1997. The WASA Approach to workflow management for scientific applications. In *Workflow Management Systems and Interoperability*, A. Dogac, L. Kalinichenko, M. T. Ozsu and A. Sheth, Eds. NATO ASI Series Vol. 164, Springer, Berlin. 145–164.

VOSSEN, G., AND WESKE, M. 1999. The WASA2 object-oriented workflow management system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM. 587–589.

WAINER, J., WESKE, M., VOSSEN, G., AND MEDEIROS, C. M. B. 1996. Scientific workflow systems. In *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions* [Online Proceedings] (May). Athens, GA.

WINFIELD, A. J. 1998. A Virtual Laboratory Notebook for simulation models. In *Proceedings of the Pacific Symposium on Biocomputing '98* (Jan.). Maui, HI. 177–88.

WOODRUFF, A. G., AND STONEBRAKER, M. 1997. Supporting fine-grained data lineage in a database visualization environment. In *Proceedings of the 13th International Conference on Data Engineering (ICDE '97)* (April). Birmingham, UK. IEEE Computer Society Press. 91–102.

WORKFLOW MANAGEMENT COALITION. 1999a. Interface 1: Process Definition Interchange—Process Model. WfMC Standard WfMC-TC-1016-P

v1.1, Workflow Management Coalition. Available at: http://www.wfmc.org/standards/docs.htm.

WORKFLOW MANAGEMENT COALITION. 1999b. Interface 1: Process Definition Interchange—Q&A and Examples. WfMC Standard WfMC-TC-1016-X v1.1, Workflow Management Coalition. Available at: http://www.wfmc.org/standards/docs.htm.

WORKFLOW MANAGEMENT COALITION. 2001. Workflow Process Definition Interface—XML Process Definition Language (XPDL). WfMC Standard WFMC-TC-1025, Workflow Management Coalition. Available at: http://www. wfmc.org/standards/docs.htm.

ZHAO, J., GOBLE, C., GREENWOOD, M., WROE, C., AND STEVENS, R. 2003. Annotating, linking and browsing provenance logs for e-Science. In *Proceedings of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data* [Online proceedings] (Oct.). Sanibel Island, FL.

Received September 2003; revised August 2004; accepted January 2005