J Data Semant (2012) 1:19–30
DOI 10.1007/s13740-012-0004-y

EDITORIAL

Scientific Workflow, Provenance, and Data Modeling Challenges and Approaches

Shawn Bowers

Received: 5 March 2012 / Accepted: 6 March 2012 / Published online: 11 April 2012

Abstract Semantic modeling approaches (e.g., conceptual models, controlled vocabularies, and ontologies) are increasingly being adopted to help address a number of challenges in scientific data management. While semantic information has played a considerable role within bioinformatics, semantic technologies can similarly benefit a wide range of scientific disciplines. Here we focus on three main areas where modeling and semantics are playing an increasingly important role: scientific workflows, scientific data provenance, and observational data management. Applications of these areas span a number of disciplines and provide both challenges and new opportunities for conceptual modeling research and development. We provide a brief overview of each area, discuss the role that modeling plays within each, and present current research opportunities.

Keywords Conceptual modeling · Semantics · Scientific workflows · Provenance and Observational data

1 Introduction

Scientists carrying out their research today face many difficult and challenging data management problems. These challenges are due in part to the changing nature of scientific research, which is increasingly based on large-scale data analysis. In this article, we broadly survey three informatics research areas that are being studied to help scientists overcome a number of data management challenges: scientific workflows, scientific data provenance, and observational data semantics.

S. Bowers (✉)
Department of Computer Science, Gonzaga University,
Spokane, WA, USA
e-mail: bowers@gonzaga.edu

The scientific workflow community is developing workflow systems to help scientists implement and execute often complex analyses involving various types of data management and computational tools. Traditionally, these types of analyses are implemented by scientists using either scripting languages (such as bash, Perl, or Python), which require fairly sophisticated computational skills, or by manually calling and storing intermediate results, which requires a considerable amount of time organizing data files and managing the overall analysis process [45]. Alternatively, scientific workflow systems help scientists by providing higher-level modeling approaches for explicitly specifying analyses as well as by providing a number of generic services for optimizing workflow execution, visualizing workflows and workflow results, managing intermediate data products, and storing the details of past workflow runs.

The ability to automatically track and record each step and dependency of a workflow run is often referred to as workflow "provenance", which has become a significant area of research within the scientific data management community. Using provenance information, scientists are able to more easily validate their analytical results and determine causal dependencies among data products. Provenance information is also crucial to the scientific process, both for data reuse and for determining reproducibility of scientific results [50].

While scientific workflows emphasize the processes used in modeling scientific analyses, the ability to describe and annotate data products themselves is crucial for interpreting, sharing, and reusing scientific data. Many scientific disciplines today leverage existing data collected by other researchers to perform analyses at broad geographic, temporal, and biological scales. These data are used to provide a better understanding of wide-ranging phenomena in which a single individual or research group cannot realistically collect the data needed to carry out a study. Classic examples

in earth and environmental science include examining the effects of nitrogen treatments across North American grasslands [56], and studying how changing environmental conditions affect bird migratory patterns [62]. These types of studies often require access to hundreds of data sets collected by independent research groups over many years. However, observational data collected in this way is inherently heterogeneous and often is not accompanied by rich enough semantic information to automatically perform the types of integration needed to combine and use the data in broader scientific analyses. Recently, a number of efforts have been developed to help provide richer conceptual models for observational data with the goal of helping researchers deal with complex, heterogeneous observational data.

We briefly summarize the work being carried out in each of these three areas as well as opportunities for future research. We focus primarily on modeling challenges within scientific workflows, provenance, and observational data, with the hope that more researchers in the conceptual and semantic modeling communities will find opportunities to apply their own research work to these three important areas in scientific data management. The rest of this article is organized as follows. In Sects. 2 through 4 we describe scientific workflows, data provenance, and observational data approaches, respectively. Each section provides an overview together with a number of modeling challenges and directions for future research. Section 5 concludes with a brief summary of the challenges presented as well as issues and future work concerning the integration of scientific workflows, provenance, and data semantics.

2 Scientific Workflows

Science is an exploratory process involving cycles of observation, hypothesis formation, and experiment design and execution. Scientific knowledge discovery is increasingly driven by data analysis and computational methods, which is due in part to technological advances in data collection instrumentation and the availability of commodity clusters for data-intensive and high-performance scientific computing. Scientific workflows can be applied during various phases of the larger scientific process to help researchers model and automate their computational experiments, data analyses, and data management tasks. The results from applying scientific workflows can yield new data and insights and thus may lead to a better understanding of or modifications to a given hypothesis or experiment outcome.

A scientific workflow is a high-level description of the processes used to carry out (often complex) computational and analytical experiments. Scientific workflows are modeled as directed graphs consisting of task nodes and dataflow or control-flow edges denoting execution dependencies among tasks. Each task within a scientific workflow represents a specific computational step (e.g., within a simulation study), data analysis step, or data management step. Common types of tasks within scientific workflows include scientific data acquisition, integration, reduction, visualization, and publication (e.g., in a shared database). These steps are sometimes modeled through composite tasks that serve as subworkflows defined over lower-level tasks. A scientific workflow is executed by a scientific workflow system. During workflow execution, workflow systems generally schedule tasks to be invoked according to the dataflow and control-flow edges of the workflow. Many scientific workflow systems allow scientific workflows to be designed visually using various forms of block diagrams (as one example, see Fig. 1).

2.1 Scientific Workflow Systems

Workflow-based approaches have been studied in the database community (e.g., [35,36,46,67,70]), within business process modeling (e.g., [1,2]), and within systems implementing problem-solving environments (e.g., [33,53]). More recently, a number of systems have been developed to provide explicit support for scientific workflow modeling and execution; for general surveys on scientific workflow systems see [25,28,73]. Examples of widely used scientific workflow systems include Taverna [49], Kepler [40], VisTrails [9], Triana [43], Pegasus [26], and Galaxy [32], among many others. Systems have also been developed to help scientists publish, share, and reuse workflow descriptions, e.g., MyExperiment [24]. A number of web-based portal applications have also been developed (e.g., [4]) that provide higher-level user interfaces for accessing and executing scientific workflows which are stored and managed using distributed server-side resources.

While scientific workflow systems largely focus on automating the execution of scientific workflows, they typically provide additional features including support for assisting users in workflow design and workflow composition, workflow execution monitoring, workflow optimization (e.g., exploiting dataflow for parallel execution), fault-tolerant execution, and the ability to record and store runtime execution information (i.e., provenance). These additional features also distinguish scientific workflow systems from more traditional script-based approaches for automating scientific data analysis (e.g., using shell, Python, or R scripts) in which such functionality is usually not provided.

As mentioned above, scientific workflows are often created within a visual editing environment in which workflows are represented as directed graphs that link atomic and/or composite tasks together. Atomic tasks can include native functions of the workflow system, but often correspond to invocations of local applications, remote (web) services, or functions within other languages (e.g., Matlab or R func-

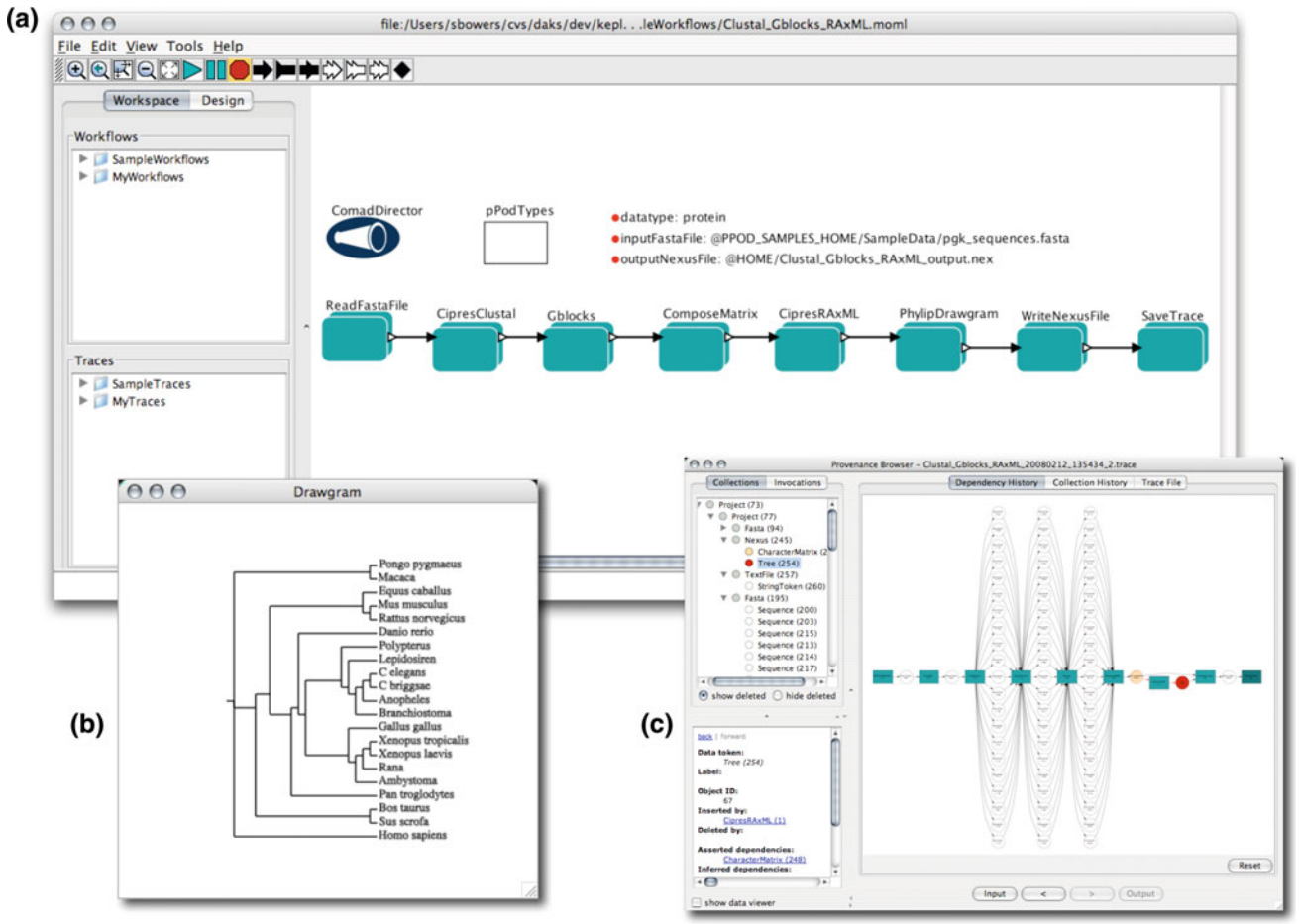

Fig. 1 Example scientific workflow in the Kepler system: **a** the user interface for creating, editing, and executing scientific workflows; **b** a visual representation of the data product (a phylogenetic tree) computed by a workflow run; and **c** a viewer for navigating the data provenance (lineage) captured in an execution trace. This workflow (which is modeled within the COMAD framework) uses a combination of local and remote (web) services to perform multiple sequence alignment and phylogenetic tree inference on input DNA sequences

tions). Scientific workflows differ from conventional programming, however, in that the workflows are often more coarse grained and involve wiring together pre-existing components and specialized algorithms. As an example, Fig. 1a shows a simple workflow defined within the Kepler workflow system. This workflow uses a combination of local applications and remote web services to perform sequence alignment and phylogenetic tree inference on input DNA sequences. The phylogenetic tree created by the analysis is also visualized within the workflow system, which is specified via a task in the workflow. The result of this particular workflow on an example data set is shown in Fig. 1b.

Scientific workflow systems span virtually all areas of the natural (and more recently social) sciences. Early applications of scientific workflow systems largely focused on supporting bioinformatics analyses, however, the spectrum of disciplines employing scientific workflow systems is much wider today, and includes application areas such as particle physics, chemistry, neurosciences, ecology, geosciences, oceanography, atmospheric sciences, astronomy and cosmology, and the social sciences, among others [29,64].

2.2 Scientific Workflow Modeling and Semantics

There is currently no standard language for modeling scientific workflows, and standards from related communities (e.g., BPEL) have not found widespread adoption in the scientific workflow community. For example, workflow systems that primarily aim at supporting workflows within high-performance computing and grid environments represent scientific workflows as simple directed acyclic graphs (DAGs) of jobs, which are then scheduled on a computational grid or compute cluster according to the implied task dependencies of the workflow. In this model of computation, each task is executed only once per workflow run and task scheduling amounts to finding a topological sort for the partial order implied by the DAG (or in some cases exploiting the partial ordering of jobs to execute tasks in parallel).

In addition, data are typically represented at a coarse-grained level in the form of files, which are implicitly passed between jobs (i.e., data passing is not explicitly modeled as part of the workflow). Alternatively, a number of workflow systems (including Kepler) support cyclic workflow graphs, in which a cycle denotes looping, e.g., for modeling while, do-while, and fixed point computations. One side effect of allowing cycles within a workflow graph is that tasks within a workflow must be executed (i.e., invoked) multiple times.

A number of other more sophisticated models of computation consider tasks as independent and continuously executing processes that can receive and send multiple data items per workflow run. Scientific workflow systems that support these types of computation models can be used for pipelined and stream-based data processing. In these cases, multiple data tokens (which wrap data or references to data) can be supplied as input to a workflow, and individual tasks are invoked over subsets of these tokens, where each task invocation can produce multiple data tokens, and so on. Examples include tasks for computing sliding-window aggregates (e.g., "running averages") and for independently applying a specific function over individual input tokens (similar in spirit to the `map` higher-order function used in functional programming languages). Depending on the system, data tokens may represent atomic data items such as integers and strings, more complex structures such as tuples or (nested) data collections, or references to external files or data sets.

Similar to business workflows, formal approaches such as Petri nets can be used to describe scientific workflow execution semantics. However, as mentioned above, the models of computation of many scientific workflow systems can exhibit both task and pipeline parallelism where token order is important. Thus, the majority of scientific workflow systems are based on dataflow semantics [29], and specifically employ dataflow computation models similar to Kahn Process Networks [38]. While most workflow systems support one type of computation model, the Kepler system allows workflow designers to select their desired model of computation (referred to as a "director") to use for a particular workflow. The two most frequently used models of computation for Kepler workflows are synchronous data flow (SDF) [37] and process networks (PN). In both SDF and PN, dependencies between tasks denote buffered data communication channels. In SDF, a serial schedule is determined prior to workflow execution based on the token consumption–production rates declared for each task. In PN, tasks are executed concurrently. Each task is assigned a separate thread within PN, and tasks are invoked as they receive tokens. The PN model can be implemented as either a data-driven ("eager") model, where data is pushed through the workflow, or as a demand-driven ("lazy") model, where data is "pulled" through the workflow (allowing users, e.g., to step through data results one output at a time). For valid SDF workflows, maximum channel buffer sizes can be guaranteed and computed prior to workflow execution, whereas in PN, buffers grow dynamically (and for some workflows may require unbounded memory) [38].

The Kepler workflow in Fig. 1 uses an extended PN model that provides explicit data and workflow modeling support for managing heterogeneous nested data collections. In particular, the collection-oriented modeling and design (COMAD) director in Fig. 1 specifies that workflow components work on a continuous, XML-like data stream. Each task in the workflow can be configured to work over certain (tagged) data collections—referred to as the component's scope. Tasks in COMAD are automatically invoked over their target collections, and all data outside of the task's scope are automatically forwarded to downstream tasks. Tasks can copy, remove, and add new data and collections within their scope. All changes to the scope items are also forwarded to downstream tasks. COMAD provides a number of benefits to workflow designers including workflows that are often more linear than equivalent workflows modeled using PN or SDF [45], which in turn makes relatively complex workflows easier to comprehend and evolve over time. The Taverna [49] system also provides similar support for nested data collections, which are modeled as nested lists of tokens. In Taverna, tasks can be configured to work over input lists using a set of pre-defined patterns (e.g., applying the task iteratively over each element of a list or sublist).

2.3 Scientific Workflow Modeling Challenges

While considerable progress has been made in modeling support for scientific workflows, a number of challenges and opportunities for future work remain. Here we briefly summarize some current challenges and prior work in these areas.

Many scientific data analyses are complex and can involve hundreds of independent steps, large amounts of heterogeneous data, and multiple data derivation paths. Because of this complexity, workflows can be difficult to design, especially for non-technical scientific users. One reason for this stems from the need to incorporate, within a single workflow, analysis tools that were not originally designed to work together, and where, e.g., no single, standard data model is used among the tools. One approach for dealing with this problem has been to use so-called "shims" within workflows [55], which act as small processing steps for transforming data into the formats needed by each tool. However, the use of shims not only makes workflows considerably harder to design, but can also make workflows considerably harder to understand and reuse. Approaches like COMAD and the use of list execution patterns in Taverna can help by allowing tasks to work over a configurable scope, however, they do not inherently deal with data conversion and transformation. Another approach that has been explored is the use of ontolo-

gies within workflow systems for annotating the inputs and outputs of workflow steps with terms drawn from domain-specific ontologies [10,14,72]. These annotations allow steps to be more easily discovered and can be used by the workflow system to make suggestions for possible shims to be used between steps [13,31].

However, a more general solution would be to explicitly make workflow systems "data-model-aware", e.g., by adding explicit data-modeling capabilities to the workflow system itself (similar to the newer approach of "artifact-centric" business workflows [34] being studied within the business process management community). For instance, much of the complexity of heterogeneous data formats could be alleviated by allowing more general conceptual models of data to be defined together with mapping tools to underlying formats, and explicitly defining workflow tasks together with their expected formats. In this way, the workflow system could automatically convert data into and out of the formats needed by analysis steps within a workflow, thus removing the need to explicitly model data transformations within a workflow. In general, scientific workflow systems today provide limited data modeling support, and this lack of support introduces various types of complexities into workflow design.

Another design challenge stems from the need to introduce control-flow constructs (such as conditional execution) into scientific workflows, e.g., to describe the desired behavior of the workflow based on runtime information or for handling faults. Because most workflow systems are based on the dataflow model of computation, control-flow is often difficult to describe and can lead to many additional (and low-level) tasks and dependencies when modeled using only dataflow constructs. This added level of complexity also make workflows difficult to modify, extend, and reuse [54]. One approach for dealing with this issue is to integrate behavioral aspects of workflows by combining dataflow with control-flow models. For instance, in [54], Kepler's support for composite tasks and explicit computation models was used to embed finite-state transducers within subworkflows to model reactive behavior for fault tolerance. Other approaches have also been developed, including the use of special-purpose control-flow components [22] as well as developing so-called "adaptive" workflows [66]. While these approaches help to simplify workflow models by decoupling control-flow modeling from dataflow modeling, these approaches have not been widely adopted within current scientific workflow systems, and more work is still needed to formalize and to integrate the expressivity of business workflow modeling languages with the more traditional dataflow languages used in scientific workflow systems.

Finally, a number of systems provide mechanisms for defining so-called "abstract" workflows [25,29,31], which add a conceptual modeling layer over concrete (i.e., executable) workflows. Abstract workflows typically take the form of templates (e.g., [30,54]), which specify the general set of high-level steps to be carried out for a particular type of analysis problem. Some systems (e.g., Pegasus and the Wings extension [31]) provide support for taking an abstract workflow and automatically suggestion possible concrete, executable versions based on task metadata. Conceptual workflow modeling can provide a number of benefits to workflow designers by allowing workflows to be developed at a higher level of abstraction (e.g., without having to worry about implementation details), for workflow discovery (e.g., to find all workflows that implement a particular template), and for workflow reuse and interoperability. However, there are currently no standard or general-purpose approaches for defining and formalizing the notion of abstract dataflow models. Given the complexity of data, control-flow, and dataflow aspects of most scientific workflows, formal conceptual workflow modeling frameworks that integrate each of these aspects would have the potential to significantly help users design and implement complex scientific analyses.

3 Scientific Workflow Provenance

Most scientific workflow systems record data provenance information during workflow execution, which typically consists of the tasks that were executed as part of the run (together, e.g., with the parameter settings of each task) and the data input to and output by each task invocation [23]. The provenance of a particular run is often referred to as a workflow trace. In addition to the inputs and outputs of tasks, some systems also record or later infer fine-grained data dependencies, which state causal relations between task inputs and outputs. For example, consider a simple task that maps a function f over each input value in a list $[x_1, x_2, \ldots, x_n]$ to produce a new output list $[y_1, y_2, \ldots, y_n]$ such that for each $1 \leq i \leq n$, $y_i = f(x_i)$. In this case, a dependency $x_i \leftarrow y_i$ exists between each x_i and y_i value stating that y_i was derived explicitly from x_i (as opposed to other values in the list).

Workflow provenance information represents important metadata that can be used by scientists to help interpret, validate, debug, and reproduce scientific analyses. The provenance of scientific data is also used to help determine data quality, e.g., to help decide whether to reuse a particular data result, as well as an aid in the process of data integration, e.g., by providing detailed information on the methods used to derive data.

Figure 2 shows a number of standard views of workflow provenance information from an example run of an image manipulation workflow (for magnetic resonance imaging) [51]. The top of Fig. 2 shows a standard data dependency graph in which nodes represent data tokens and edges denote data dependencies. Each dependency edge is labeled

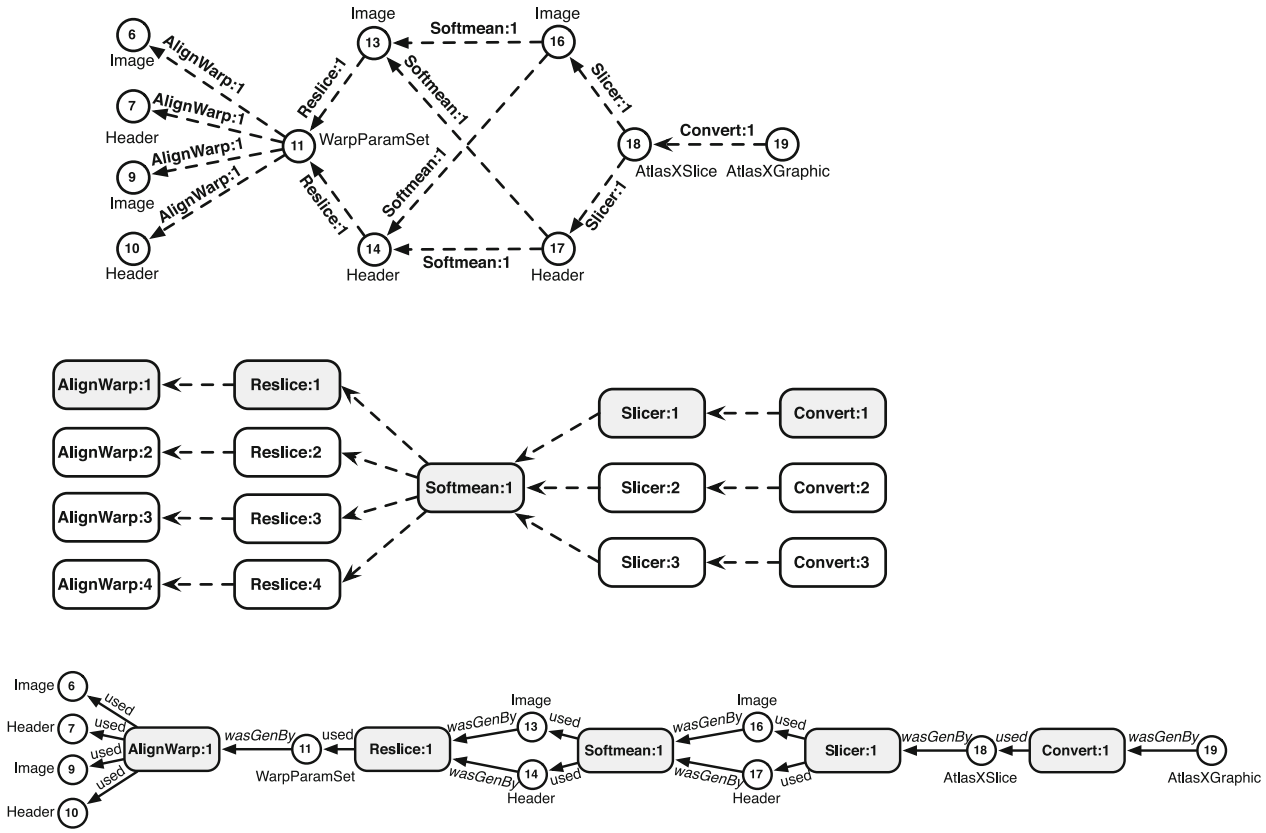

Fig. 2 Three different provenance views of an example run of the first provenance challenge workflow: **a** a portion of the fine-grained dependency graph corresponding to the first invocation of each workflow task; **b** an invocation dependency graph for the complete workflow run; and **c** `used` and `wasGeneratedBy` edges from OPM for the first invocation of the run

by its corresponding task invocation. For instance, the label `Softmean:1` between node 13 and 16 states that the Image represented by token 16 was generated by the first invocation of the `Softmean` workflow task, the Image represented by token 13 was input to this invocation, and Image 16 was derived directly from Image 13. The middle of Fig. 2 shows a standard invocation dependency graph in which nodes represent task invocations and edges denote dependencies between tasks. For example, the edge between the `Softmean:1` task invocation and the `Reslice:1` task invocation states that an output of `Reslice:1` was input to `Softmean:1`, and this output was used to derive an output of `Softmean:1`. The invocation graph of Fig. 2 shows the complete set of invocations of the trace, whereas the dependency graph shows only the data dependencies introduced by the first invocation of each workflow task in the run.

Provenance graphs provide a natural representation for provenance information and in general, the information captured within provenance graphs can be used in a variety of ways. For example, workflow developers can leverage provenance to verify that a workflow design executes properly by examining the inputs and outputs of task invocations, or by ensuring that all input data is used to derive at least one corresponding output data product. Similarly, scientists can use provenance graphs to help understand how workflow results were derived by determining the data, tasks, and invocation parameters that contributed to a particular data product. Provenance information can also be used to discover data and workflows, e.g., by enabling users to find data, possibly across workflow runs, derived from certain input data (in case data is revised or found to contain errors), or to find workflows used to produce data of a certain type. A number of scientific workflow and provenance-based systems support queries over provenance information. Examples of common types of queries include [16,51]:

- What were the inputs and/or outputs of this run?
- What were the inputs, outputs, and/or parameters of this invocation?
- What data and/or invocations were used to derive this output?
- What data was derived from this input?
- Were specific data and invocation dependencies satisfied?

Queries are often expressed using either standard query languages such as SQL, XQuery, or SPARQL, or via declarative query languages designed specifically for provenance (e.g., [7,11,39,60,61]).

3.1 Provenance Modeling Approaches

The Open Provenance Model (OPM) [50] attempts to standardize a number of basic relationships found within the models of provenance employed by a number of workflow and provenance systems. OPM defines four types of entities: Processes, which are similar to task invocations; Artifacts, which are similar to tokens; Agents, which direct processes; and Accounts, which act as containers for representing different views of the same set of provenance events. OPM also defines a number of provenance relationships, including `used` to denote that an artifact was used by a process, `wasGeneratedBy` to denote that an artifact was produced by a process, `wasDerivedFrom` to denote artifact dependency relationships, and `wasTriggeredBy` to denote process dependencies. The bottom of Fig. 2 shows a portion of an OPM provenance graph (containing `used` and `wasGeneratedBy` edges) for the dependency graph shown at the top of the figure. Kepler, Taverna, VisTrails, Pegasus, and a number of other workflow and provenance systems have been extended to support OPM, allowing traces captured within each system to be represented and exchanged using the OPM standard. A series of "provenance challenges"[1] were created to compare provenance support across workflow systems as well as to evaluate the effectiveness of using OPM as a standard interchange approach for workflow provenance.

While OPM provides a baseline standard for provenance information, most models of provenance employed within workflow systems are driven by the underlying model of computation supported. For instance, both Taverna and COMAD provide additional provenance modeling constructs for representing the provenance of nested data collections. Other systems provide considerably more detail than the relationships supported by OPM [23], e.g., token read and write timestamps [3], task parameter information [7], data and task annotations [48], support for composite tasks [68], and workflow specifications and modifications [9]. In general, provenance models are largely driven by the set of runtime observables that can be recorded and are available within the workflow system (and in particular, the corresponding model of computation). More complex models of computation (e.g., involving streaming, concurrency, distributed scheduling, etc.) require additional provenance modeling constructs to faithfully capture the details of a workflow run.

A number of ontology-based approaches have also been created for representing workflow provenance information. The Provenir ontology [58,59], e.g., introduces OWL-DL classes for representing common provenance terms and relationships. An advantage of this approach is that the ontology can be easily extended with additional terms for describing domain-specific information related to data and tasks. For instance, Provenir is extended in [58] to support a sensor network application in which domain-specific classes are introduced to describe the sensors used for collecting input data sets as well as the corresponding workflows used to process this data. In Janus [48], domain-specific ontologies are used to annotate the more traditional "domain agnostic" provenance representation of Taverna [47]. Janus also extends the Provenir ontology to support annotation of provenance graphs. Finally, the W3C Provenance Working Group[2] is currently standardizing a generic provenance model (i.e., one that is not solely designed for representing scientific workflow provenance), but that is largely based on the OPM and Provenir approaches. The W3C provenance working group is also currently developing a provenance ontology in OWL-DL for expressing provenance information within the context of the web.

3.2 Provenance Modeling Challenges

As mentioned above, one challenge in modeling scientific workflow provenance is finding an appropriate model of provenance for a given model of computation. As workflow languages become more expressive (e.g., adding control-flow constructs to dataflow computation models), provenance representations must also be supplemented with new modeling features to support these changes. Within the context of provenance information, tasks are often characterized as being either "white box" or "black box" components. White box components refer to tasks that can be statically analyzed such as those expressed using SQL queries. In this case, fine-grained dependencies can often be inferred directly from the definition of the task [19]. Alternatively, black-box components refer to tasks where the underlying implementation is not visible to the workflow system, implying that fine-grained data dependencies must be explicitly declared, e.g., by task or workflow developers. For initial work on combining white-box and black-box provenance approaches, see [5]. Alternatively, many workflow systems allow additional levels of specification of tasks in addition to the underlying task implementation itself. The ability to specify a task's scope within the COMAD computation model is one such example. In these cases, we can think of the task as a "gray box" that mixes both white-box features (through task wrapping and customization) and black-box features (since the under-

[1] http://twiki.ipaw.info/bin/view/Challenge/.

[2] http://www.w3.org/2011/prov/wiki/Main_Page.

lying code is not directly analyzable). The ability to define gray-box specifications is important not only for provenance, but also for making workflows easier to develop for users. However, suitable and formal languages and approaches for modeling gray-box features is still largely unexplored.

Another often overlooked challenge within scientific workflow provenance centers on distinguishing different types of dependencies. For instance, OPM along with many other provenance models consider only one generic type of data dependency relationship. Data dependencies can be established for a number of reasons, e.g., a dependency relationship may represent a derivation (where the value of an input was used to compute a new value), a copy operation (the output was a copy of the input), control-flow (the input was used to trigger the task), and so on.

Another challenge related to workflow provenance concerns application support, e.g., for visualization and browsing, incorporating provenance information into data quality metrics, provenance analytics and mining (e.g., to determine workflow patterns and similarities), incorporating provenance information into workflow design, and ensuring privacy when publishing provenance information. A number of approaches have been developed to help users view and navigate relevant portions of provenance graphs generated from large-scale workflows. Kepler, e.g., uses a simple browser for displaying provenance graphs (see Fig. 1c). The Zoom*UserViews approach [11] extends these approaches by allowing users to select relevant tasks from which the system automatically generates a set of well-defined abstract composite tasks to simplify provenance navigation and query. Other similar approaches (e.g., [6]) have also been developed to support user-defined summarization and interactive browsing of provenance large provenance graphs.

4 Observational Data

A considerable amount of scientific knowledge is derived either directly or indirectly from relatively simple, underlying measurements linked to real-world phenomena. These measurements are often recorded and stored in observational data sets that are analyzed by researchers using a variety of tools and methodologies. Many fields increasingly use observational data from *multiple* disciplines (genetics, biology, geology, hydrology, sociology, etc.) to tackle broader and more complex scientific questions. Within the earth and environmental sciences, e.g., cross-disciplinary data is necessary to investigate complex environmental issues at broad geographic and temporal scales. Carrying out such studies requires the integration and synthesis of observational data from multiple research efforts [8,27]. Effectively using observational data, however, can be extremely challenging for researchers due to its inherent structural and semantic heterogeneity. This in turn makes it difficult to find relevant data sets, interpret data sets once found, and then combine data collected by other researchers for analysis.

The heterogeneity of observational data is a result of a number of factors, including: (1) most observational data are collected by individuals, institutions, or scientific communities through independent (i.e., uncoordinated) research projects; (2) the structure of observational data is often chosen based on collection methods (e.g., to make data easier to record "in the field") or the format requirements of analysis tools, as opposed to standard representations and schemas for data; and (3) the terms and concepts used to label data are not standardized, both within and across scientific disciplines and research groups [41]. However, as research becomes increasingly cross-disciplinary, there is a growing need in a number of communities to provide richer metadata and data representation approaches to help unify access to observational data sets.

4.1 Observational Data Modeling Approaches

The need for a more uniform mechanism to describe observational data has led to a number of proposals for observational data models [20,21,63] and ontologies [42,44,57,71]. Many of these provide approaches tailored to their specific scientific domains of interest for describing data and storing observational data. Other efforts are aimed at developing more generic and extensible approaches for modeling observational data semantics. Figure 3 shows one example of a generic model for describing observational data [15,18].

The model of Fig. 3 defines constructs to describe and (depending on the implementation) store observational data. An *observation* is made of an *entity* (e.g., biological organisms, geographic locations, or environmental features, among others) and primarily serves to group a set of measurements together to form a single "*observation event*". A *measurement* assigns a value to a *characteristic* of the observed entity (e.g., the height of an entity). Measurements also include *standards* (e.g., units) for relating values across measurements, and can also specify additional information including collection protocols, methods, precision, and accuracy (not all of which are shown in Fig. 3). An observation (event) can occur within the *context* of zero or more other observations. Context can be viewed as a form of dependency, e.g., an observation of a tree specimen may have been made within a specific geographic location, and the geographic location provides important information for interpreting and comparing different measured values. In this case, by establishing a context relationship between the tree and location observations, the measured values of the location are assumed to be constant with respect to the measurements of the tree (i.e., the tree measurements are dependent on the location measurements). Context forms a *transitive*

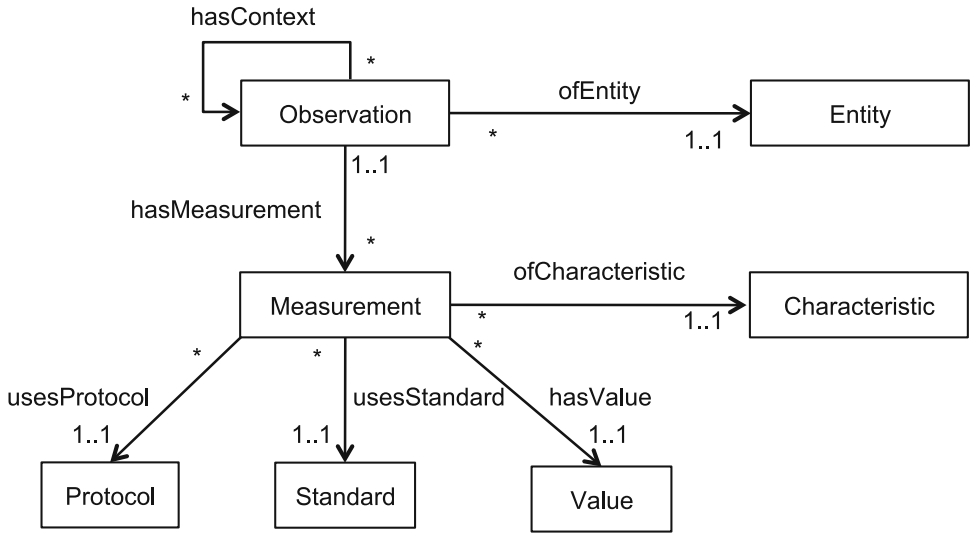

Fig. 3 Main observational modeling constructs defined in the Extensible Observation Ontology (OBOE): **a** observation (events) are recorded for entities and are composed of zero or more measurements; **b** measurements assert recorded values of observed entities and include a measurement protocol, a measurement standard (e.g., units), the measured value, and the characteristic of the entity measurement (e.g., height, mass, etc.); and **c** the contextual information associated with an observation event (such as temporal and spatial references) are represented through explicit context relations

relationship among observations. When describing data sets using the model of Fig. 3, domain-specific entity, characteristic, and standard classes are typically used, e.g., by defining subclasses of the classes of Fig. 3.

Once data are described using an observational model, it becomes considerably easier to use heterogeneous observational data sets. In particular, the models provide uniform views of the data (in terms of observation and measurement structures) that can be exploited for data-discovery queries (e.g., to find all data sets that contain observations and measurements of specific types [18]), data analysis operations (e.g., to aggregate and summarize data sets [12]), and data set transformation and integration (since the structural heterogeneity of underlying data formats is removed).

The basic concepts described in Fig. 3 are present in a number of existing observational data models including the ISO Observations and Measurements (O&M) model developed by the Open Geospatial Consortium [20], the CUAHSI ODM model [63], the Extensible Observation Ontology (OBOE) [42], the Entity-Quality (EQ) ontology model [52], and the SWEET ontology [57], among others. These approaches largely differ in terms of the technology and scientific domains they support, e.g., some models are designed specifically for relational database systems [21,63], others focus on higher-level representations (e.g., in UML) without targeting any specific implementation [20], and others use description-logic approaches by focusing on OWL-based representations and semantic-web technologies [69]. Another major difference in the approaches concerns the use of context information. For example, both O&M and OBOE provide explicit support for defining a wide range of observation context information, while others allow only certain dimensions to be defined (such as fields for specifying the location and time of the observation). The different models also adopt specific data description approaches. For instance, EQ and OBOE provide frameworks for supporting data annotation in which existing data sets are stored in their native formats but annotated with concepts from the respective ontologies. Alternatively, ODM and current implementations of O&M require data to be stored explicitly using specific relational or XML schemas, respectively.

4.2 Observational Data Modeling Challenges

While developing and using common observational models can provide significant benefits in terms of reusing and analyzing heterogeneous data sets, current approaches are still relatively new. While many existing tools for exploiting observational models focus on data interoperability (e.g., where data in one database can be exchanged using the common observational model with other systems), relatively few approaches have been developed to exploit common structures for data discovery and integration [12]. One area of future research that can make significant impact on the scientific community is the development of robust tools for (semi-)automatically combining distinct observational data sets into a single, integrated data set. For example, in [65] an approach based on possible-world semantics is described for automatically integrating biodiversity data sets. Scientists spend a considerable amount of effort manually integrating observational data, and tools and techniques to automate this process which have the potential to save researchers

considerable time and effort. Similarly, expressive and general-purpose data discovery and query languages as well as efficient implementations (e.g., for large-scale observational data sets) are also important areas of future research.

Another area of future research is to extend scientific workflow systems with support for observational models, which could provide a number of benefits for researchers. There are various ways in which the integration could occur, e.g., using a common observational data model for describing and representing data could help deal with some of the "shim" problems within workflow systems, and observation and measurement types could also be used to further describe input and output requirements of workflow tasks. Similarly, incorporating provenance information into current observational models could provide additional metadata that is often crucial for interpreting and integrating data. Observational models often provide limited support for defining detailed protocols and methods for data collection and use, which could be aided by the process modeling capabilities of scientific workflows. This information could also help constrain and guide workflow selection and configuration. Finally, leveraging observational models to design data collection and storage requirements of research studies could lead to standardized efforts for data representation across research groups. For example, in [21], researchers select the observations and measurements of interest prior to developing data collection forms, which are automatically generated based on the set of abstract observation types desired as part of the study.

5 Summary

This article has given a broad overview of three different but related research areas in scientific data management, namely, scientific workflows, data provenance, and observational data semantics, with an emphasis on the modeling approaches and challenges within each. The scientific data management community has made a number of advances within each of these areas, however, many open problems and opportunities for future research and development remain. For instance, scientific workflow modeling approaches are largely based on dataflow models, but there is a significant need to extend these with support for defining various control-flow aspects. Another major issue is adding data modeling support to workflow design, which is largely non-existent in current workflow systems. Similarly, provenance systems are largely based on a relatively simple model of provenance, whereas workflow systems often contain a number of more advanced features such as structured data and different types of dependencies that should be modeled to obtain more accurate lineage information. Finally, both scientific workflow and provenance systems lack robust support for detailed semantic information about data, especially for observational data that often contain rich semantic context that is essential for data reuse, data analysis, and data integration. While each individual area has a number of modeling challenges, more research into how to effectively combine these three areas is also needed.

In [17], one possible approach for combining data, workflow, and provenance management is proposed based on a simple conceptual model for capturing "project histories". In this work, scientific data associated with a research project is organized by a scientist using hierarchical folder-like structures. Users can add new data sources into their project history as well as data products produced by scientific workflow runs. Each project history stores detailed provenance information about its data products, which can be viewed and used to access associated workflows and workflow runs (via a "run browser"). Dependencies between data products are also explicitly captured within a project history, and users can select and rerun workflows with existing and new data products. More recently, a number of systems have adopted similar types of functionality, e.g., the Galaxy system provides similar capabilities for bioinformatics research [32].

However, many opportunities exist for new research into how to extend and combine scientific workflow modeling, data provenance, and data semantics to provide effective data management tools for scientific researchers. In particular, adopting and extending abstraction mechanisms provided by conceptual and semantic modeling to challenges in scientific data management is a promising area of future research, which has the potential to significantly enhance the ability of scientists to manage the inherent complexity of scientific data and data analysis needed to advance scientific research today.

Acknowledgments This work was supported in part by NSF grants #0743429, #0753144, and #1118088. The author would especially like to thank Bertram Ludäscher, Timothy McPhillips, Manish Kumar Anand, Mark Schildhauer, and Matthew Jones whose collaborations over many years in scientific workflows, provenance, and observational data semantics were essential for the ideas presented in this article.

References

1. van der Aalst W (1998) The application of petri nets to workflow management. J Circuits Syst Comput 8(1):21–66
2. van der Aalst W, van Kees H (2004) Workflow management: models, methods, and systems. The MIT Press, Cambridge
3. Altintas I, Barney O, Jaeger-Frank E (2006) Provenance collection support in the Kepler scientific workflow system. In: International provenance and annotation workshop (IPAW), pp 118–132
4. Altintas I, Lin AW, Chen J, Churas C, Gujral M, Sun S, Li W, Manansala R, Sedova M, Grethe JS, Ellisman MH (2010) Camera 2.0: a data-centric metagenomics community infrastructure driven by scientific workflows. In: IEEE World Congress on Services, pp 352–359

5. Amsterdamer Y, Davidson SB, Deutch D, Milo T, Stoyanovich J, Tannen V (2011) Putting lipstick on pig: enabling database-style workflow provenance. PVLDB 5(4):346–357
6. Anand MK, Bowers S, Ludäscher B (2009) A navigation model for exploring scientific workflow provenance graphs. In: Proceedings of the workshop on workflows in support of large-scale science (WORKS)
7. Anand MK, Bowers S, McPhillips TM, Ludäscher B (2009) Exploring scientific workflow provenance using hybrid queries over nested data and lineage graphs. In: SSDBM, pp 237–254
8. Andelman S, Bowles C, Willig M, Waide R (2004) Understanding environmental complexity through a distributed knowledge network. BioSciences 54(3):2400–2246
9. Bavoil L, Callahan S, Scheidegger C, Vo H, Crossno P, Silva C, Freire J (2005) Vistrails: enabling interactive multiple-view visualizations. In: IEEE visualization, pp 135–142
10. Belhajjame K, Wolstencroft K, Corcho Ó, Oinn T, Tanoh F, Williams A, Goble CA (2008) Metadata management in the taverna workflow system. In: IEEE international symposium on cluster computing and the grid (CCGRID), pp 651–656
11. Biton O, Boulakia SC, Davidson SB, Hara CS (2008) Querying and managing provenance through user views in scientific workflows. In: ICDE, pp 1072–1081
12. Bowers S, Kudo J, Cao H, Schildhauer MP (2010) ObsDB: a system for uniformly storing and querying heterogeneous observational data. In: eScience, pp 261–268
13. Bowers S, Ludäscher B (2004) An ontology-driven framework for data transformation in scientific workflows. In: International workshop on data integration in the life sciences (DILS), pp 1–16
14. Bowers S, Ludäscher B (2005) Actor-oriented design of scientific workflows. In: International conference on conceptual modeling (ER), pp 369–384
15. Bowers S, Madin JS, Schildhauer MP (2008) A conceptual modeling framework for expressing observational data semantics. In: ER, pp 41–54
16. Bowers S, McPhillips TM, Ludäscher B, Cohen S, Davidson SB (2006) A model for user-oriented data provenance in pipelined scientific workflows. In: International workshop on provenance and annotation (IPAW), pp 133–147
17. Bowers S, McPhillips TM, Wu M, Ludäscher B (2007) Project histories: managing data provenance across collection-oriented scientific workflow runs. In: International workshop on data integration in the life sciences (DILS), pp 122–138
18. Cao H, Bowers S, Schildhauer MP (2011) Approaches for semantically annotating and discovering scientific observational data. In: International conference on database and expert systems applications (DEXA), pp 526–541
19. Cheney J, Chiticariu L, Tan WC (2009) Provenance in databases: why, how, and where. Found Trends Databases 1(4):379–474
20. Cox S (2011) Observations and measurements v2.0—XML implementation. Tech Rep 10-025r1, OGC
21. Cushing JB, Nadkarni N, Finch M, Fiala A, Murphy-Hill ER, Delcambre LML, Maier D (2007) Component-based end-user database design for ecologists. J Intell Inf Syst 29(1):7–24
22. Damevski K, Khan A, Parker S (2008) Scientific workflows and components: together at last. In: Proceedings of the workshop on component-based high-performance computing (CBHPC)
23. Davidson SB, Freire J (2008) Provenance and scientific workflows: challenges and opportunities. In: SIGMOD conference, pp 1345–1350
24. De Roure D, Goble C, Stevens R (2009) The design and realisation of the myExperiment virtual research environment for social sharing of workflows. Future Gener Comput Syst 25:561–567
25. Deelman E, Gannon D, Shields MS, Taylor I (2009) Workflows and e-science: an overview of workflow system features and capabilities. Future Gener Comput Syst 25(5):528–540
26. Deelman E, Singh G, Su MH, Blythe J, Gil Y, Kesselman C, Mehta G, Vahi K, Berriman GB, Good J, Laity AC, Jacob JC, Katz DS (2005) Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Sci Program 13(3):219–237
27. Ellison A, Osterweil L, Hadley J, Wise A, Boose E, Clarke L, Foster D, Hanson A, Jensen D, Kuzeja P, Riseman E, Schultz H (2006) Analytic webs support the synthesis of ecological datasets. Ecology 87:1345–1358
28. Fox GC, Gannon D (eds) (2006) Concurrency and computation: practice and experience. Special issue: Workflow in grid systems, vol 18(10). Wiley, Chichester
29. Gil Y, Deelman W, Ellisman W, Fahringer T, Fox G, Gannon D, Goble C, Livny M, Moreau L, Myers J (2007) Examining the challenges of scientific workflows. Computer 40(12):24–32
30. Gil Y, Groth PT, Ratnakar V, Fritz C (2009) Expressive reusable workflow templates. In: International conference on e-science, pp 344–351
31. Gil Y, Ratnakar V, Kim J, González-Calero PA, Groth PT, Moody J, Deelman E (2011) Wings: intelligent workflow-based design of computational experiments. IEEE Intell Syst 26(1):62–72
32. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11(8):R86
33. Houstis E, Gallopoulos E, Bramley R, Rice J (1997) Problem-solving environments for computational science. IEEE Comput Sci Eng 4(3):18–21
34. Hull R (2008) Artifact-centric business process models: brief survey of research results and challenges. In: OTM conferences, pp 1152–1163
35. Ioannidis YE, Livny M (1989) Moose: modeling objects in a simulation environment. In: Ritter GX (ed) IFIP congress. North Holland, pp 821–826
36. Ioannidis YE, Livny M, Gupta S, Ponnekanti N (1996) ZOO: a desktop experiment management environment. In: Vijayaraman TM, Buchmann AP, Mohan C, Sarda NL (eds) Proceedings of international conference on very large data bases (VLDB), pp 274–285
37. Lee EA, Messerschmitt DG (1987) Synchronous data flow. Proc IEEE 75(9):1235–1245
38. Lee EA, Parks TM (1995) Dataflow process networks. Proc IEEE 83(5):773–799
39. Lim C, Lu S, Chebotko A, Fotouhi F (2011) OPQL: A first OPM-level query language for scientific workflow provenance. In: IEEE international conference on services computing (SCC), pp 136–143
40. Ludäscher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones MB, Lee EA, Tao J, Zhao Y (2006) Scientific workflow management and the Kepler system. Concurr Comput Pract Exper 18(10)
41. Madin J, Bowers S, Schildhauer M, Jones M (2008) Advancing ecological research with ontologies. Trends Ecol Evol 23(3):159–168
42. Madin J, Bowers S, Schildhauer M, Krivov S, Pennington D, Villa F (2006) An ontology for describing and synthesizing ecological observation data. Ecol Inform 2:279–296
43. Majithia S, Shields M, Taylor I, Wang I (2004) Triana: a graphical web service composition and execution toolkit. In: Proceedings of the IEEE international conference on web services (ICWS). IEEE Computer Society
44. McGuinness D, Fox P, Cinquini L, West P, Garcia J, Benedict J, Middleton D (2007) The virtual solar-terrestrial observatory: a deployed semantic web application case study for scientific research. In: AAAI, pp 1730–1737
45. McPhillips TM, Bowers S, Zinn D, Ludäscher B (2009) Scientific workflow design for mere mortals. Future Gener Comput Syst 25(5):541–551
46. Medeiros CB, Vossen G, Weske M (1995) WASA: A workflow-based architecture to support scientific database applications. In:

Database and expert systems application (DEXA). Springer LNCS 978, pp 574–583
47. Missier P, Paton NW, Belhajjame K (2010) Fine-grained and efficient lineage querying of collection-based workflow provenance. In: EDBT, pp 299–310
48. Missier P, Sahoo SS, Zhao J, Goble CA, Sheth AP (2010) Janus: From workflows to semantic provenance and linked open data. In: International provenance and annotation workshop (IPAW), pp 129–141
49. Missier P, Soiland-Reyes S, Owen S, Tan W, Nenadic A, Dunlop I, Williams A, Oinn T, Goble CA (2010) Taverna, reloaded. In: SSDBM, pp 471–481
50. Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth PT, Kwasnikowska N, Miles S, Missier P, Myers J, Plale B, Simmhan Y, Stephan EG, den Bussche JV (2011) The open provenance model core specification (v1.1). Future Gener Comput Syst 27(6):743–756
51. Moreau L, Ludäscher B, Altintas I, Barga RS, Bowers S, Callahan SP, Jr. GC, Clifford B, Cohen S, Boulakia SC, Davidson SB, Deelman E, Digiampietri LA, Foster IT, Freire J, Frew J, Futrelle J, Gibson T, Gil Y, Goble CA, Golbeck J, Groth PT, Holland DA, Jiang S, Kim J, Koop D, Krenek A, McPhillips TM, Mehta G, Miles S, Metzger D, Munroe S, Myers J, Plale B, Podhorszki N, Ratnakar V, Santos E, Scheidegger CE, Schuchardt K, Seltzer MI, Simmhan YL, Silva CT, Slaughter P, Stephan EG, Stevens R, Turi D, Vo HT, Wilde M, Zhao J, Zhao Y (2008) Special issue: The first provenance challenge. Concurr Comput Pract Exp 20(5):409–418
52. Mungall C (2007) Representing phenotypes in owl. In: Proceedings of the workshop on OWL: experiences and directions (OWLED)
53. Nakagawa AS (1994) LIMS: implementation and management. The Royal Society of Chemistry, Thomas Graham House, The Science Park, Cambridge CB4 4WF
54. Ngu AHH, Bowers S, Haasch N, McPhillips TM, Critchlow T (2008) Flexible scientific workflow modeling using frames, templates, and dynamic embedding. In: SSDBM, pp 566–572
55. Oinn T, Greenwood M, Addis M, Alpdemir MN, Ferris J, Glover K, Goble C, Goderis A, Hull D, Marvin D, Li P, Lord P, Pocock MR, Senger M, Stevens R, Wipat A, Wroe C (2006) Taverna: lessons in creating a workflow environment for the life sciences. Concurr Comput Pract Exp 18(10)
56. Pennings S, Clark C, Cleland E, Collins S, Gough L, Gross K, Milchunas D, Suding K (2005) Do individual plant species show predictable responses to nitrogen addition across multiple experiments? Oikos 110(3):547–555
57. Raskin R (2004) Enabling semantic interoperability for earth science data. http://sweet.jpl.nasa.gov
58. Sahoo SS, Barga RS, Sheth AP, Thirunarayan K, Hitzler P (2009) PrOM: a semantic web framework for provenance management in science. Tech. Rep. KNOESIS-TR-2009, Kno.e.sis Center
59. Sahoo SS, Sheth AP, Henson CA (2008) Semantic provenance for escience: managing the deluge of scientific data. IEEE Internet Comput 12(4):46–54
60. Scheidegger CE, Koop D, Santos E, Vo HT, Callahan SP, Freire J, Silva CT (2008) Tackling the provenance challenge one layer at a time. Concurr Comput Pract Exp 20(5):473–483
61. Simmhan YL, Plale B, Gannon D (2008) Query capabilities of the karma provenance framework. Concurr Comput Pract Exp 20(5):441–451
62. Sorokina D, Caruana R, Riedewald M, Hochachka W, Kelling S (2009) Detecting and interpreting variable interactions in observational ornithology data. In: ICDM workshops, pp 64–69
63. Tarboton D, Horsburgh J, Maidment D (2007) CUAHSI community observations data model (ODM), version 1.0 design specifications. http://water.usu.edu/cuahsi/odm/
64. Taylor I, Deelman E, Gannon D, Shields M (eds) (2007) Workflows for e-Science: scientific workflows for grids. Springer
65. Thau D, Bowers S, Ludäscher B (2009) Merging sets of taxonomically organized data using concept mappings under uncertainty. In: OTM conferences, pp 1103–1120
66. Tolosana-Calasanz R, Bañares JA, Rana OF, Álvarez P, Ezpeleta J, Hoheisel A (2010) Adaptive exception handling for scientific workflows. Concurr Comput Pract Exp 22(5):617–642
67. Wainer J, Weske M, Vossen G, Medeiros CB (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
68. Wang L, Lu S, Fei X, Chebotko A, Bryant HV, Ram JL (2009) Atomicity and provenance support for pipelined scientific workflows. Future Gener Comput Syst 25(5):568–576
69. Wang P, Zheng J, Fu L, Patton EW, Lebo T, Ding L, Liu Q, Luciano JS, McGuinness DL (2011) A semantic portal for next generation monitoring systems. In: International semantic web conference (ISWC), pp 253–268
70. Wiener JL, Ioannidis YE (1993) A moose and a fox can aid scientists with data management problems. In: C. Beeri, A. Ohori, D. Shasha (eds) 4th international workshop database programming languages (DBPL). Springer, pp 376–398
71. Williams R, Martinez N, Goldbeck J (2006) Ontologies for ecoinformatics. J Web Semant 4:237–242
72. Wolstencroft K, Alper P, Hull D, Wroe C, Lord PW, Stevens RD, Goble CA (2007) The myGrid ontology: bioinformatics service discovery. IJBRA 3(3):303–325
73. Yu J, Buyya R (2005) A taxonomy of scientific workflow systems for grid computing. SIGMOD Record 34(5)