Incorporating Semantics in Scientific Workflow Authoring (Demonstration Paper)

Chad Berkley, Shawn Bowers, Matthew B. Jones, Bertram Ludäscher, Mark Schildhauer, Jing Tao
{berkley, jones, schild, tao}@nceas.ucsb.edu, {bowers, ludaescher}@ucdavis.edu

Abstract. *The tools used to analyze scientific data are often distinct from those used to archive, retrieve, and query data. A scientific workflow environment, however, allows one to seamlessly combine these functions within the same application. This increase in capability is accompanied by an increase in complexity, especially in workflow tools like Kepler, which target multiple science domains including ecology, geology, oceanography, physics, and biology. To overcome this complexity, we have developed semantically-driven user-interface components that are customized at run-time using domain-specific ontologies. One such subsystem in Kepler uses domain-specific ontologies to customize the presentation of analytical components and data for use by scientists building workflows. Kepler also provides for semantically-enabled queries for components, which can significantly increase efficiency in workflow authoring tasks. In this demonstration, we show how ontologies can be used for user-interface customization and more. In particular, we show our recent ontology-driven extensions for workflow authoring in Kepler. These extensions include our advances in: (1) automating data-integration and service-composition tasks, (2) the use of semantic annotations to verify that workflows are semantically meaningful, and (3) the ability to search for contextually relevant components and data sets* in situ, *i.e., as a user is designing a scientific workflow. The utility of these extensions will be shown within the context of developing species biodiversity analyses within Kepler.*

1. Introduction

Scientific workflow systems have traditionally been stand-alone applications designed for a specific domain. For example, physicists, geologists, ecologists, and oceanographers typically use their own applications (e.g., a set of ”MATLAB” scripts) for creating and executing scientific workflows. The Science Environment for Ecological Knowledge (SEEK) [SEEK] project is developing a powerful, cross-domain scientific-workflow authoring environment that allows scientists to design and execute novel workflows. The need for such a tool has been recognized in other scientific domains, and so SEEK has teamed up with several other projects, including GEON [GEON], SDM [SDM], EOL [EOL] and ROADNet [ROADNET] to produce *Kepler* [KEPLER].

Scientific workflow systems such as Kepler provide scientists with a number of benefits. In particular, they provide an integrated environment in which scientists can design, communicate, and execute their analytical processes. They typically incorporate a variety of functions for end-to-end workflow execution and management, including data query, retrieval, and archiving tools. And, they provide a mechanism to help scientists recreate previous analyses (thus allowing workflows to serve as a form of metadata) and provide an opportunity for workflows (and data) to be reused to form novel and extended analyses.

A major challenge for Kepler is to effectively support users from different scientific disciplines, while maintaining both generic support for scientific workflows and enabling cross-domain data and workflow reuse. Instead of creating complex interfaces and tools for each domain, we desire the capability to provide domain-specific customization. We believe that ontologies can be used not only to formalize domain knowledge, but also to support creation of customized user interfaces, thus facilitating cross-domain interaction.

As part of SEEK (and in collaboration with the other projects previously noted), we are actively engaging scientists to develop ontologies, with the goal of having a rich repository of domain-specific terminologies and cross-linkages among them. Along with this effort, we are also developing a suite of ontology-based tools [BLL04, BL04, BTWL04] to allow scientists to more easily browse, query, integrate, and compose relevant cross-discipline datasets and services. This demonstration will highlight these ontology-enabled tools and their implementation within Kepler.

2. Scientific Workflows and Kepler

A scientific workflow is an executable description of a scientific process. In particular, a scientific workflow records each inline process required to take input data and produce a meaningful output product. Scientific workflows are similar to business-process workflows but have several properties uncommon to the business environment. For example, scientific workflows often operate on large, complex, and heterogeneous data. They can be computationally intensive, and can produce complex derived data products that may be archived for use in re-parameterized runs or other workflows. Moreover, unlike business workflows that are often event-flow driven, scientific workflows are generally data-flow driven (i.e., execution is based on the flow of data as opposed to triggered events).

In Kepler, scientific workflows bring together data and services, possibly created by groups or individuals

unknown to each other. Moreover, the workflow applications written in Kepler encompass a wide variety of scientific domains, sub-domains, and specialties. By making these data and services broadly accessible and comprehensible way, Kepler facilitates cross-domain investigations and interdisciplinary research.

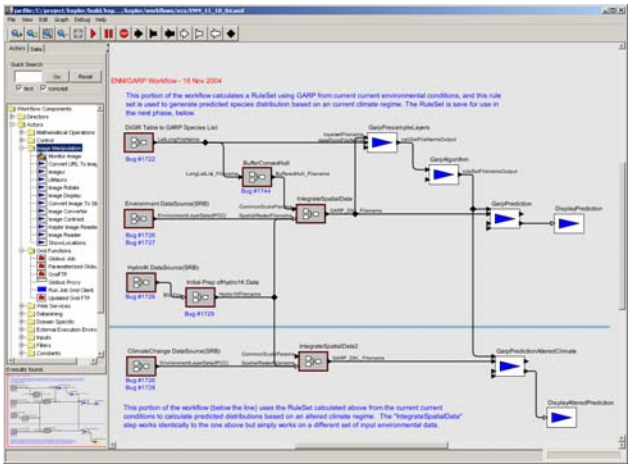


Figure 1. The Kepler scientific workflow environment.

Within Kepler, scientific workflows are authored in a graphical, drag-and-drop manner. Services contain typed ports that can be connected to other services or data sources. Ports can have simple atomic types such as *integer* and *string* as well as more complex structures, including arbitrarily nested array and record types. As a workflow is executing, data passes between ports via tokens that can be readily manipulated to meet the differing syntactic needs of other services. Data produced by a scientific workflow can be displayed graphically at run time, or written to disk for later use.

An example scientific workflow within Kepler is shown in Figure 1. This workflow executes an ecological niche model. Shown is a high-level view of the workflow in which components on the left (darker rectangles) are sub-workflows, embedded in this higher-level workflow. The components shown with a triangle are atomic services that directly call Java code to perform their functions. The specific workflow displayed in Figure 1 uses a genetic algorithm to do species range predictions.

A feature of modeling this prediction as a workflow is that a scientist can easily change underlying services, e.g., use a different statistical prediction algorithm (service) or a different visualization/presentation service.

3. Conceptual Challenges in Scientific Workflows

Because Kepler is a powerful and flexible workflow system with a diverse set of users, a number of conceptual-modeling challenges arise. Our goal is to allow users with different backgrounds and varying levels of computing expertise to create new scientific workflows with a minimum amount of difficulty. We highlight below the main difficulties we wish to address.

Supporting high-level conceptual models. Most scientists have a high-level conceptual model of their workflows. If asked, a scientist can typically write down the steps involved in taking raw data and producing their desired output fairly quickly. However, when this conceptual model becomes formalized into an executable scientific workflow, a large number of low-level technical details arise. Details such as file access, network protocols, dataset schemas, service input and output typing, execution models (e.g., tuple-at-a-time versus table-at-a-time dataflow), and configuration parameters tend to obscure the high-level conceptual model of the workflow, making it hard to compare it with existing workflows and reuse it in new settings. We would like to effectively capture the high-level aspects of a workflow, while also preserving but often hiding the underlying technical details.

Basic contextual metadata. A general lack of contextual metadata with respect to data and services is problematic for users (e.g., those who are trying to find new and relevant datasets and services). As an example, a service titled "interpolator" might give one the impression that it provides a generic interpolatation operation over arbitrary datasets when in fact, the service was written to interpolate spatial grid data. Additionally, the same component could simply have been named "int," obscuring the functionality of the service even though those familiar with the particular workflow know that "int" means "spatial data grid interpolate". We face the challenge of making these services generically comprehensible and accessible.

Schema and service-type semantics. Scientific data integration can be a complex and time-consuming process. Scientific data is highly heterogeneous, laden with structural, schematic, and semantic differences. Today, scientific-data integration is typically performed by hand and requires significant "meta" information. Service composition similarly requires considerable contextual information describing structure (to manage heterogeneity in input and output types) and semantics (the kind of things consumed or produced by a service).

4. Using Semantics in Workflow Authoring

Robust metadata is required to meet the challenges involved in enabling domain scientists to create, run, and share scientific workflows. Several communities continue to have grass-roots organizations that deal with the collection and storage of syntactic metadata. The Knowledge Network for Biocomplexity [KNB] serves the ecological community with the Ecological Metadata Language (EML) [EML] and associated metadata repositories [JBBS01]. Other relevant metadata standards

for Kepler include FGDC [FGDC] and Dublin Core [DC], to name a few.

While standards such as EML may provide some support for semantic metadata (e.g., using a "keyword" field), this information is typically not sufficiently formalized for general use in an automated environment. Most current metadata standards for services also fail to include such formal semantics, including the Web-Service Description Language (WSDL) [WSDL] and the Modeling Markup Language (MoML) [MOML].[1]

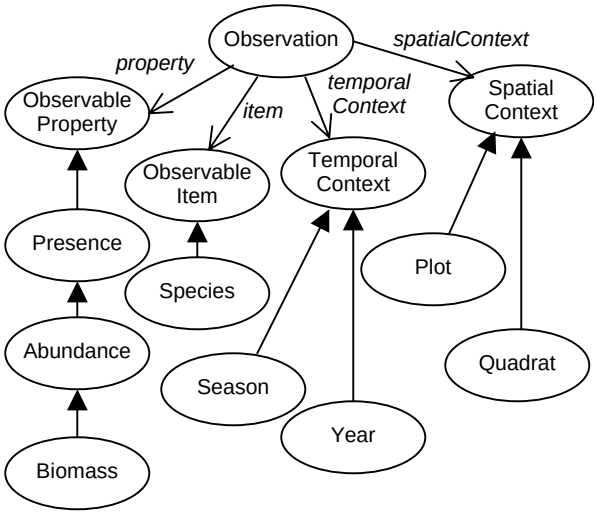


Figure 2: A simplified SEEK ontology.

Kepler has adopted the OWL web ontology language [OWL] (more specifically, OWL-DL) as the primary language for capturing domain-specific terminologies used in semantic metadata. Our approach is to leverage OWL-DL ontologies and *semantic annotations* (described below) of data and services within Kepler to capture rich and possibly complex semantic metadata. A fragment of the SEEK "measurement" ontology is shown graphically in Figure 2.

To address the conceptual challenges discussed in the previous section, we have developed the following features, which we propose to demonstrate. Each of these features leverages the domain ontologies being developed within SEEK and the other Kepler projects.

Support for detailed semantic annotations. Kepler is designed to provide users with the ability to semantically register [BLL04] their dataset schemas and services (and their corresponding input and output types) using *semantic annotations.* Figure 3 gives a set of semantic annotations for the *biom* dataset containing species biomass observations. A semantic annotation defines a relationship between a service or dataset and terms in an ontology. Intuitively, semantic annotations define the "semantic type" of the resource (shown by the statements on the left of the arrows in Figure 3), and link portions of the semantic type to portions of the resource (shown on the right of the arrow in Figure 2). For example, the first annotation in Figure 3 states that tuples in the *biom* dataset denote *Observation* instances from the ontology (in Figure 2). Similarly, the second annotation states that a year value within a tuple denotes the corresponding observation's temporal context and is an instance of the *Year* ontology concept. The semantic annotation language is designed for use at different "granularities," e.g., from selecting a single concept and assigning it to a service, to prescribing a complex ontology instantiation and assigning individual structures within it to particular data values within a dataset (such as in figure 3).[2]

Dataset Schema:
biom(yr, seas, plt, qd, spp, bm)

Semantic Annotations:

x:biom	==>	*x*:Observation
x:biom[yr=*y*]	==>	*x*[*temporalContext*=*y*:Year]
x:biom[seas=*s*], *s*='W'	==>	*x*[*temporalContext*=s:Winter]
x:biom[seas=*s*], *s*='S'	==>	*x*[*temporalContext*=s:Spring]
x:biom[seas=*s*], *s*='F'	==>	*x*[*temporalContext*=s:Fall]
x:biom[plt=*p*]	==>	*x*[*spatialContext*=*p*:Plot]
x:biom[qd=*q*]	==>	*x*[*spatialContext*=*q*:Quadrat]
x:biom[spp=*s*]	==>	*x*[*item*=*s*:Species]
x:biom[bm=*b*]	==>	*x*[*property*=*b*:Biomass]

Figure 3: Example semantic annotations.

Workflow-component classification and browsing. Kepler leverages semantic annotations to provide customizable access to datasets and services. As shown in Figure 1, the panel on the left displays hierarchically arranged concepts taken from a user-selected ontology, and automatically places services within the hierarchy. This feature provides Kepler users the ability to: 1) select and configure the classification ontology, 2) view the hierarchically arranged ontology (which is computed using a description-logic classifier), and 3) see services classified according to the concept hierarchy (by matching these up through their semantic annotations). In this way, users can easily customize Kepler service presentation (similarly for datasets), and provide ontology-based browsing of data and services.

Semantic scientific-workflow analysis. Given a workflow of interconnected actors, Kepler statically checks (i.e., at design time) whether two connected

[1] An exception is the proposed OWL-S [OWL], which provides a "heavy-weight" language for defining the semantics of services.

[2] Semantic annotations in Kepler differ from other approaches by providing rich semantic descriptions that can be "superimposed" over structural types and schemas, allowing explicit connections between substructures and semantic types.

services (or data sources) are "semantically compatible" based on their semantic annotations, and notifies the user when a connection is not considered semantically well typed. This capability directly assists a user with the workflow creation process.

Ontology-directed scientific-workflow design. As large repositories of workflow components become available, finding relevant resources becomes more difficult. Given a workflow service on the Kepler canvas (the right side of Figure 1), a user can search for all "semantically compatible" resources (either datasets or services) that can be connected to the input (or output) of the service. This search can also be restricted to return resources that are both semantically and structurally compatible (using Kepler's type system).

(Semi-)Automated Integration and Composition. Scientists often reuse existing workflow components to construct new models. Such components are more often than not structurally incompatible, even though they may be semantically compatible. Our goal is to exploit semantic annotations to derive structural correspondences between input and output data types [BL04]. These correspondences often contain enough information to derive the desired data transformations, allowing scientists to state the desired component connection instead of the low-level details of how those connections should be made. Similarly, multiple datasets must often be combined (i.e., merged or integrated) to be useful as input to a workflow. In this demonstration, we will also show our recent developments for assisting Kepler users in the process of data integration [BTWL04] and service composition, leveraging semantic annotations and domain-specific ontologies.

5. Conclusions and Future Work

In our demo we will show our recent ontology-driven extensions to Kepler for workflow authoring. These extensions include (1) our advances in automating data-integration and service-composition tasks, (2) the use of semantic annotations to verify that workflows are semantically meaningful, and (3) the ability to search for contextually relevant components and data sets in situ, i.e., as a user is designing a scientific workflow. The utility of these extensions will be shown within the context of developing species biodiversity analyses within Kepler.

Acknowledgments

Kepler is based upon the Ptolemy II code base. Kepler includes contributors from SEEK [SEEK], SDM Center-SPA [SPA], Ptolemy II [PTOLEMY] and Geon [GEON]. Work supported in part by NSF ITRs 022567 (SEEK), 0225673 (GEON).

References

[BLL04] Bowers S., K. Lin, and B. Ludäscher, 2004. On integrating scientific resources through semantic registration, In Proc. of SSDBM, pp. 349-352.

[BL04] Bowers S. and B. Ludäscher, 2004. An ontology-driven framework for data transformation in scientific workflows, In Proc. of Workshop on Data Integration in the Life Sciences (DILS), LNCS, vol. 2994, pp. 1-16.

[BTWL04] Bowers S., D. Thau, R. Williams, and B. Ludascher, 2004. Data procurement for enabling scientific workflows: On exploring inter-ant parasitism, In Proc. of Workshop on Semantic Web and Databases (SWDB), LNCS, vol. 3372.

[DC] Dublin Core Metadata Initiative. http://dublincore.org.

[EML] Ecological Metadata Language. http://knb.ecoinformatics.org/software/eml/.

[EOL] Encyclopedia of Life. http://eol.sdsc.edu/.

[FGDC] Federal Geospatial Data Committee. http://www.fgdc.gov.

[SP99] Stockwell D.R.B. and D. Peters 1999. The GARP Modeling System: problems and solutions to automated spatial prediction. International Journal of Geographical Information Science 13 (2): 143-158.

[GEON] The Geosciences Network. http://www.geongrid.org

[JBBS01] Jones, M.B., C. Berkley, J. Bojilova, and M. Schildhauer, 2001. Managing Scientific Metadata, IEEE Internet Computing 5(5): 59-68.

[KEPLER] Kepler: An Extensible System for Scientific Workflows, http://kepler.ecoinformatics.org

[KNB] Knowledge Network for Biocomplexity. http://knb.ecoinformatics.org.

[LL00] E. A. Lee and S. Neuendorffer. 2000. "MoML - A Modeling Markup Language in XML, Version 0.4," Technical Memorandum UCB/ERL M00/12, University of California, Berkeley, CA 94720, March 14, 2000. http://ptolemy.eecs.berkeley.edu/publications/papers/00/moml/.

[OWL] Web Ontology Language. http://www.w3.org/TR/owl-features/.

[PTOLEMY] Ptolemy II, http://ptolemy.eecs.berkeley.edu/ptolemyII/.

[ROADNET] Real-time Observatories, Applications and Data Management Network. http://roadnet.ucsd.edu.

[SDM] Scientific Data Management Center. http://sdm.lbl.gov/sdmcenter/.

[SEEK] Science Environment for Ecological Knowledge. http://seek.ecoinformatics.org.

[WSDL] Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl.