To Appear in AAAI Fall Symposium on Proactive Assistant Agents, Arlington, VA, November 2010.

Assisting Scientists with Complex Data Analysis Tasks through Semantic Workflows

Yolanda Gil, Varun Ratnakar, and Christian Fritz

Information Sciences Institute
University of Southern California
gil@isi.edu, varunr@isi.edu, fritz@isi.edu

Abstract

To assist scientists in data analysis tasks, we have developed semantic workflow representations that support automatic constraint propagation and reasoning algorithms to manage constraints among the individual workflow steps. Semantic constraints can be used to represent requirements of input datasets as well as best practices for the method represented in a workflow. We demonstrate how the Wings workflow system uses semantic workflows to assist users in creating workflows while validating that the workflows comply with the requirements of the software components and datasets. Wings reasons over semantic workflow representations that consist of both a traditional dataflow graph as well as a network of constraints on the data and components of the workflow.

Introduction

While there has been much research over the years on large-scale scientific computing, scientists still lack appropriate support to face the enormous complexity of data analysis tasks. Although they may be intimately familiar with the driving scientific problem, they are often presented with analysis tools that require expansive knowledge of statistical and analytic methods as well. Because this knowledge is often not readily available, scientists must invest effort in understanding which kinds of analytical techniques and tools are most appropriate for their problem. This amounts to reading myriads of technical publications and software documentation for a vast array of analytical commercial and open source software. Based on this understanding, they must set up software that enables the execution of individual steps involved in the overall analysis pipeline. In executing these steps, they must keep track of a large number of constraints imposed by each analytic tool to ensure that the analysis is valid.

As a result, carrying out data analysis processes remains a highly manual, time consuming, individualized, and error prone process. Validation of the results is also time consuming as there is no mechanism to offer reassurance that the software and data are correctly combined, particularly when analyses are carried out by less experienced researchers. Reproducibility is a great challenge, as many details of an original analysis are buried in notebooks and often are only known to the research assistants that set up and run the software. Finally, it is hard for researchers to always use state-of-the-art techniques, as new analytical tools and methods are constantly appearing in the field and each requires a significant time investment to understand, setup, and use.

Workflows have emerged as a key component of scientific infrastructure that enables the representation of data analyses in a declarative manner, which explicitly capture the dataflow across components [Gil et al 07a]. Workflows have been used to manage complex scientific applications in areas as diverse as biomedical imaging, genomics, astronomy, and geophysics among others [Taylor et al 07; Deelman et al 09]. Workflows capture an end-to-end analysis composed of individual analytic steps as a dependency graph that indicates dataflow links as well as control flow links among steps. Workflow systems are of great value to scientists because they automate many aspects of the execution of complex scientific applications. However, they have no capabilities to exploit semantic metadata or constraints, and therefore workflow systems cannot reason about the type of analysis being done to the data. As a result, the kind of assistance that they can provide is very limited.

To assist scientists in creating valid workflows that are appropriate for their datasets, workflow systems should be able to represent the semantics of the data analysis application that they are managing. Based on these representations, they should support automatic constraint propagation and reasoning algorithms to automatically manage constraints among the individual workflow steps. Semantic workflow systems could then assist users in a variety of ways, by validating their use of workflows for complex analyses, initializing the workflow settings, and finding published datasets to support their ongoing analysis [Gil 2009].

This paper presents our work on semantic workflows to assist scientists in setting up and configuring workflows. We show this in the context of the Wings workflow system

[Gil et al 09a; Gil et al 09b; Gil et al 10]. The paper begins with some background on the state of the art in what is available today for a scientist in terms of data analysis and workflow systems. Next, we introduce the Wings framework for semantic workflows. We then present the capabilities to assist scientists that Wings offers.

Workflow Systems for Scientific Data Analysis

A variety of systems are available that provide users with a repository of useful data analysis components that can be readily executed. Some also allow them to stitch together software pipelines, typically with a linear data flow across components. However, the assistance to the user is very limited, as they all focus on the convenience of a single entry point for running software and have no facilities to assist the user with selecting or configuring workflows. We give here an overview of their capabilities.

Some systems are essentially software libraries that include implementations of many algorithms and statistical routines that are commonly used in a community. Examples of these systems include R and Bioconductor [Chambers et al 98; Gentleman et al 05]. Individual routines can be invoked from a command line, and are used in sequence, where the user selects a command and can use its output as the input to the next one. The system keeps a history of the commands, so commands can be used again in the future. These command sequences form a linear pipeline for data processing, which is a very simple kind of workflow. Because scientific datasets can be very large, some packages support execution of individual algorithms in computer clusters. However, these tools possess no capacity for more complex data dependencies among the commands, or any assistance to users with the methodology beyond textual documentation.

Other systems contain all of the above capabilities, but also capture the complete analytical method with "workflows". These systems represent data analyses in a declarative manner, which explicitly capture the dataflow across components. Examples of these systems include GenePattern [Reich et al 06] and Galaxy [Giardine et al 05]. These systems include many analytic tools that are easily accessible from a single point of entry provided in their interface. Furthermore, the analytic tools are composed into workflows that can be browsed, re-executed, and re-used by others.

Workflow systems represent more sophisticated workflow structures and manage their execution in distributed environments [Taylor et al 05]. They represent data flow and control flow dependencies among individual components. However, workflow systems tend to focus on the mechanics of assembling the software pieces and on managing the execution through remote web services or shared cyberinfrastructure resources. They do not have any capabilities to assist the scientist with science-level matters, such as what software components are appropriate for their data and how to configure them to fit the characteristics of the data. As a result, workflow composition remains a manual process where scientists get little assistance in validating the workflows that they create.

However, these human-intensive approaches are impractical for the tsunami of scientific data. In biomedical research, new techniques in imaging and next generation sequencing at very low cost will yield in the order of terabytes of data per individual and their use for personal medicine makes an analysis timeframe of several months completely unacceptable. In environmental science, large amounts of sensor data are being collected in cyberobservatory networks that will need to be processed very quickly in order to influence agency and government policies (e.g., agriculture permits, reservoir management) in time to avoid environmental catastrophes. Scientists need better guidance and assistance to create valid workflows in order to make data analysis processes more efficient and therefore more useful.

Semantic Workflows and the Wings Workflow System

Most workflow systems can only assist scientists by automating the execution of workflows and contain little knowledge about what the workflow represents. In order to provide user assistance, workflow systems need to be augmented with the ability to reason about the workflow and its constituents.

Semantic workflows represent declaratively the metadata properties of their constituents [Gil 2009]. They represent and reason about the constraints posed by each software component in the workflow (i.e., a given analytic tool) on the type of data that it can process, the kind of parameter settings that are more appropriate for those data, and the constraints and properties of the output datasets that it generates. Semantic workflow representations can support automatic constraint propagation through steps and reasoning algorithms to manage constraints among the individual workflow steps.

Wings is a semantic workflow system that uses semantic representations of workflows, software components, and data in order to assist users in generating valid execution-ready workflows [Gil et al 2009a; Gil et al 2009b; Gil et al 2010a; Gil et al 2010b]. Figure 1 illustrates a semantic workflow in Wings as shown in its user interface. This is a workflow that discretizes datasets, then creates a model to classify test data. Semantic constraints are shown at the top, the dataflow is shown graphically at the bottom. Software components are shown as rectangles, datasets and component parameters as ovals. The semantic constraints stating properties of workflow constituents such as software components and intermediate data are shown as simple triples of <object property value>. The system contains many other kinds of more complex constraints that are implemented as rules. They are not shown in this user interface, because it was designed for assistance with

workflow set up and configuration and therefore showing complex constraints would hinder usability. More complex constraints are only accessible in the Wings interface for workflow developers [Gil et al 2010a].

Semantic representations can be the basis to assist scientists in applying complex scientific analyses by guiding them to apply workflows while respecting the constraints for a valid use of the analytic tools and methods. The focus of this paper demonstrates how this is done in Wings. The rest of this section summarizes other capabilities supported by semantic workflows in the Wings framework.

Semantic representations of workflows can be used for **automated workflow generation**. Wings uses a four-step workflow generation algorithm that takes into account metadata properties of the datasets processed by the workflow as well as constraints on the components [Gil et al 2010a; Gil et al 2010b]. These properties and constraints are used to select components, datasets, and parameter values automatically from a high-level user request.

Semantic workflows can also be used to **generate metadata attributes** for all the new data products of the workflow. In Wings this is done by propagating metadata from the input data through the descriptions and constraints specified for each of the components [Kim et al 2006]. This enables detailed records of how new data products were generated by the workflow and the parameter configuration of each component, which are captured in a provenance catalog that can support repeatability of experiments [Kim et al 2008].

Semantic workflows can also support **reasoning about parallel execution of data and component collections**. Wings represents the semantic properties of data collections in a workflow template that is then elaborated into as many parallel paths according to the user configuration and parameterization of the workflow [Gil et al 2009b].

Semantic workflows allow the **search and discovery of workflows based on their properties**. Wings can search workflow catalogs that include semantic representations based on queries that describe properties of input datasets or desired properties of the workflow results [Gil et al 09a].

The Wings workflow system has an open modular design and can be easily integrated with other existing workflow systems and execution frameworks to extend them with semantic reasoning capabilities. We have integrated the Wings semantic workflow system with other user interfaces, and submitted workflows with a variety of execution engines. Wings is built on open web standards from the World Wide Web Consortium (W3C) such as the Web Ontology Language (OWL), the Resource Description Framework (RDF), and the SPARQL query language for RDF. Wings has been used in a number of application domains including geosciences [Gil et al 2007b], genomics, social network analysis, and student assessment [Ma et al 2010].

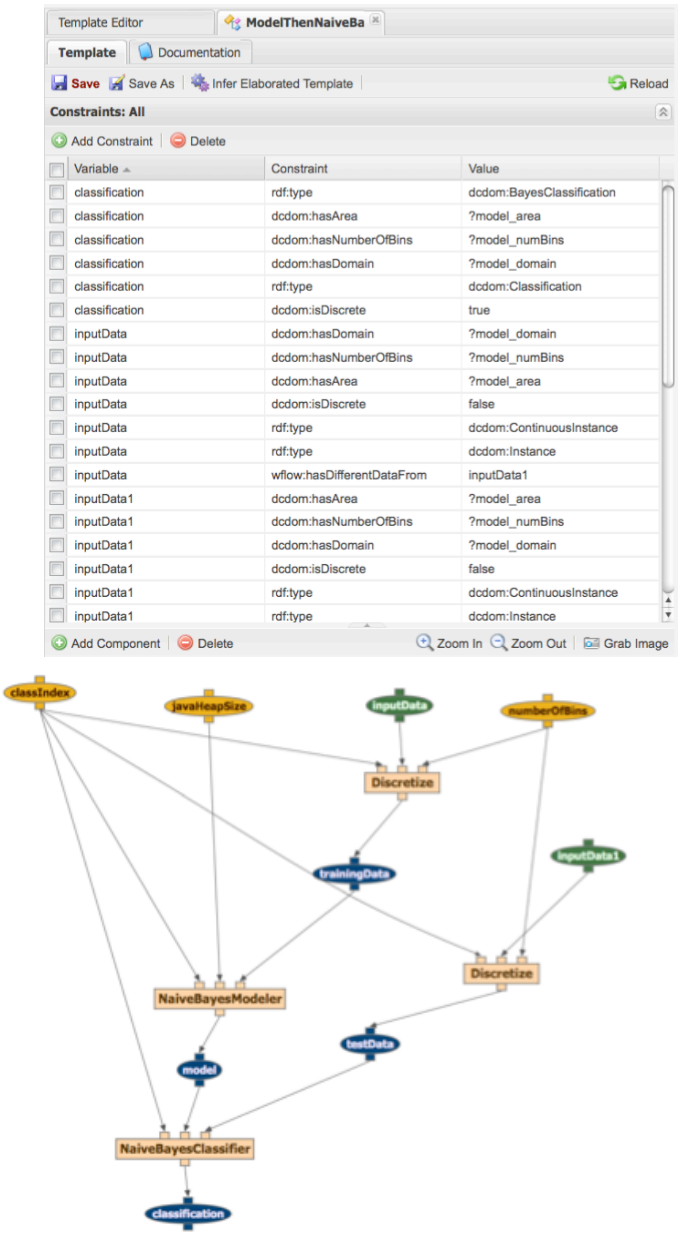


Figure 1. A workflow that discretizes datasets, then creates a model to classify test data. Semantic constraints are shown at the top, stating properties of workflow constituents such as software components and intermediate data. The dataflow is shown graphically at the bottom.

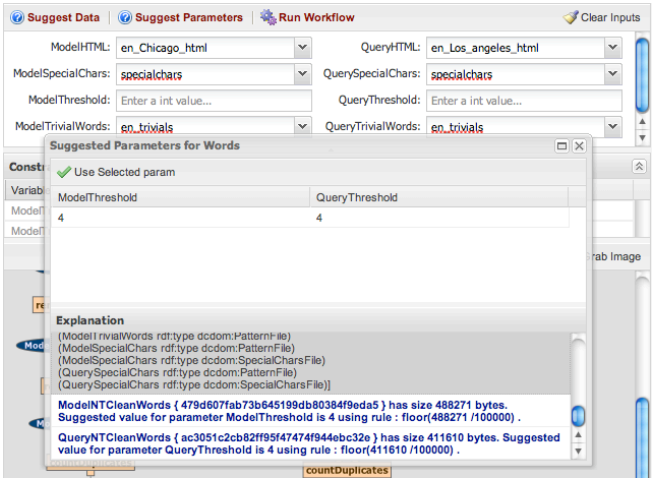
Semantic Workflows: Unique Features of Wings

Wings can represent semantic constraints of workflows and reason about them to assist a user. This section shows several scenarios that illustrate the user assistance that Wings can provide.

In this section we use illustrative examples from simple workflows to create topic models with text files. The input files can be in PDF or html, the workflow removes the markup, removes special characters (eg commas) and common words (such as "the") and counts how many times each word appears. That is the topic model, and can be compared to the models of other files. The topic model can have a minimal threshold of how many times a word must appear in order for it to be included.

Assistance to Set Up Workflow Parameters

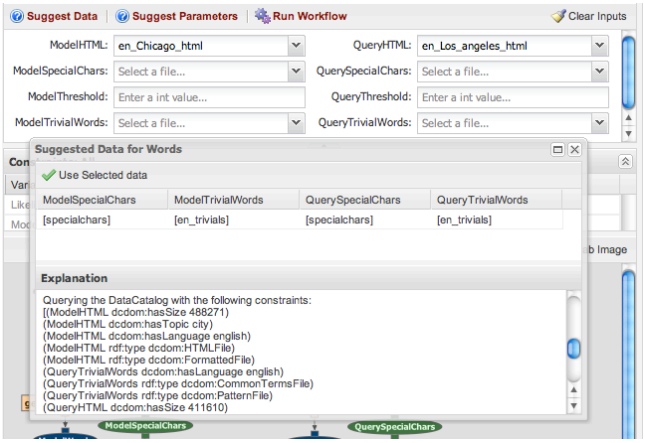
Wings can propose parameter values for a workflow once the user has selected what datasets they want to use. To do this, Wings reasons about semantic constraints that determine the best values of parameters based on metadata properties of the datasets selected. Once the user has uploaded and selected datasets for the workflow, they can ask Wings to "Suggest parameters", which results in a pop-up window with the following results:



In this case, there is a constraint that the threshold parameters should be set depending on the metadata property *size* of the input. It is important to note that these suggested values are not constant or default values, but instead are selected based on the input datasets. If the user selects a different input dataset, then the system will suggest different values for these parameters. The pop up window offers this explanation highlighted in blue. The explanation also contains other inferences that Wings has made based on propagating constraints throughout the workflow. The suggestions for parameter values often depend on these inferences, which are not part of the semantic constraints table that you see for the workflow.

Assistance to Find Relevant Datasets

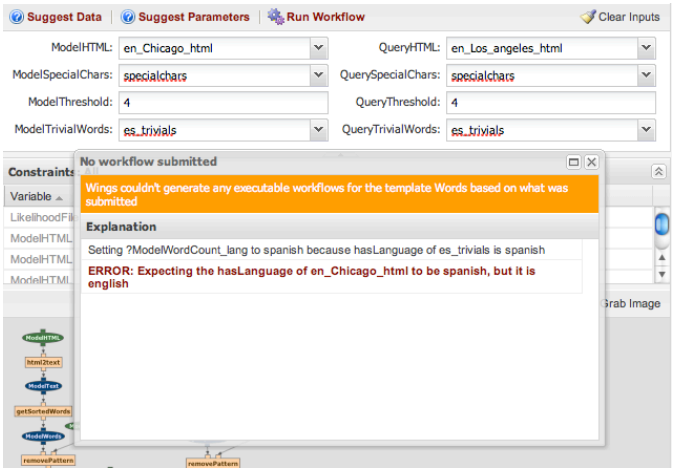
The user can ask Wings for suggestions on what datasets to use. This is needed when the user has some of the inputs but would like the system to use reference datasets that are standard or shared. An example in our workflow is the dataset of special characters. After selecting some of the inputs, the user can ask Wings to "Suggest data":



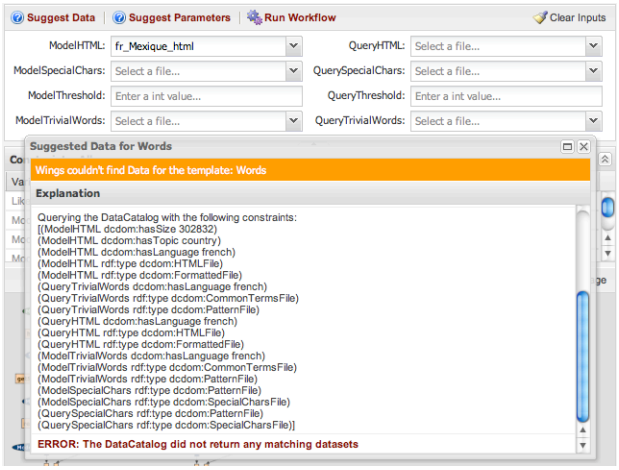
Using a constraint that states that the special characters files must have the same *language* as the input files, Wings suggests appropriate pattern files from all those available. An explanation is shown to justify why the particular datasets suggested are appropriate. This explanation consists of semantic constraints that have been propagated through the workflow by Wings and that are not part of the semantic constraints table of your workflow description.

Assistance to Validate a Workflow

If the user selects input files and parameter settings that are inconsistent with the semantic constraints that are defined in the system, Wings will detect that and let them know that the workflow is invalid. For example, we mentioned earlier a constraint that the language of the input files and the language of the pattern files has to be the same. If files with disparate languages are selected, Wings will warn that it is an invalid workflow:



Wings can also alert a user if no workflow is possible for the selected input data. In the following example, the file selected in French and there are no pattern files available in the system in that language that could be used:



Wings figures this out based on the semantic constraint for removePattern that its input pattern files have to be in the same language as the input word list files.

This validation capability is very helpful, as Wings is keeping track for the user of all the constraints that are defined in the domain and ensuring that any workflow that a user creates is valid.

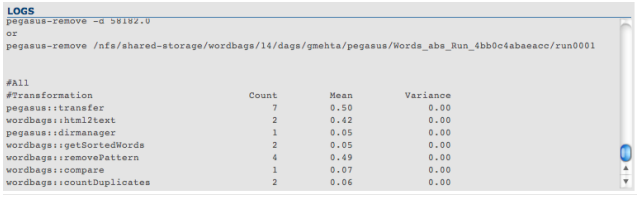
Assistance to Specialize an Abstract Workflow

In Wings, workflows can have abstract components that represent a class of executable components. This workflow has an abstract component:

The component removeMarkup is a class of components that includes two executables: html2text and pdf2text. Depending on what kind of data is input to the workflow, the appropriate executable will be selected.

Note that this workflow is similar to the workflow that we used above except it is more general, as it can take in any formatted file (i.e., either an HTML or a PDF File). The "removeMarkup" component is an abstract step, and the system will specialize it automatically as an html Markup removal step or a pdf markup removal step depending upon the type of the input that is given to it.

For example, if the user selects an html file, the execution trace shows that the html2text component was used:



Assistance to Manage the Parallel Processing of Collections of Data

Wings can reason about a collection of datasets so that it is processed in parallel by the workflow. A workflow that can process data collections is marked in the user interface by a multi-layered oval in the workflow diagram:

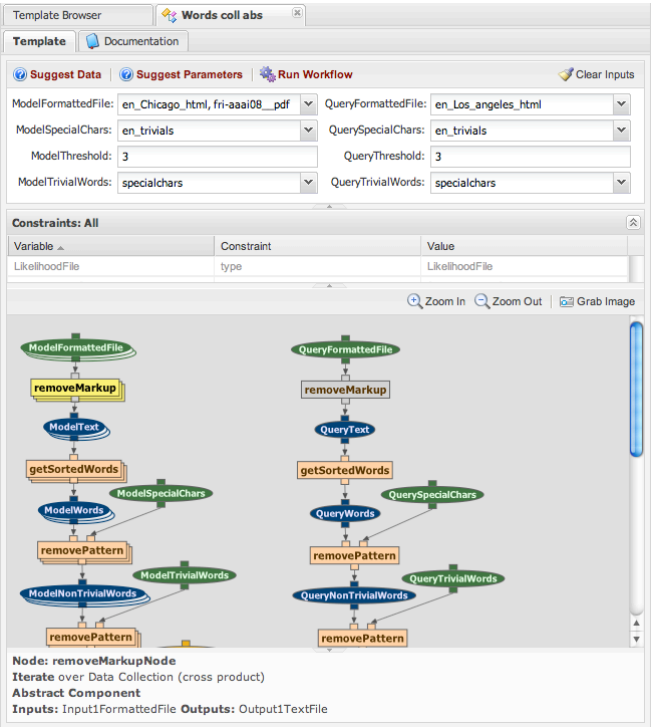
When the user selects inputs, a drop down menu allows the selection of several files at a time. At the bottom of the screen the user can see a note that it is a collection.

When the workflow is submitted for execution, Wings will create individual jobs for each of the files to be processed. In this case, the input is a one-dimensional collection, as it is a list of files. Wings can handle multi-dimensional collections of datasets, as well as collections of workflow components.

Assistance with Selection of Workflow Components

Suppose that the user wants to process a collection of files, some are in html format and some in pdf. It would be

useful if the workflow system to select the right workflow component for removing markup for each of the files according to their format. Wings can do exactly that using the following workflow:



Conclusions

Semantic workflows can be used to assist scientists in creating and validating complex data analysis tasks. the Wings workflow system uses semantic representations of workflow constraints to reason about the scientists goals and requirements. Wings can assist a scientist in setting up and validating analyses by suggesting values for parameters and by checking that the user's setup respects the constraints of the given workflow. Wings can also find datasets relevant to an analysis by reasoning about the workflow selected by the user and the goals of the analysis expressed as constraints on the workflow results.

Acknowledgements. This research was supported by the National Science foundation with grant CCF-0725332.

References

Chambers, JM. "Programming with Data." Springer Verlag, 1998.

Gentleman, R., Carey, et al. "Bioinformatics and Computational Biology Solutions Using R and Bioconductor". Springer Verlag, 2005.

Deelman, E., G. Singh, et al. "Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems". Scientific Programming, 13, 2005.

Deelman, E., Gannon, D., Shields, M., Taylor, I. "Workflows and e-Science: An overview of workflow system features and capabilities". Future Generation Computer Systems, 25(5), 2009.

Giardine B, Riemer C, et al. "Galaxy: a platform for interactive large-scale genome analysis." Genome Research 15(10):1451-5, 2005.

Gil, Y., Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox, Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers. "Examining the Challenges of Scientific Workflows," IEEE Computer, vol. 40, no. 12, pp. 24-32, December, 2007.

Gil, Y., Varun Ratnakar, Ewa Deelman, Gaurang Mehta, and Jihie Kim. "Wings for Pegasus: Creating Large-Scale Scientific Applications Using Semantic Representations of Computational Workflows," Proceedings of the 19th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), Vancouver, British Columbia, Canada, July 22-26, 2007

Gil, Y. "From Data to Knowledge to Discoveries: Scientific Workflows and Artificial Intelligence." Scientific Programming, Volume 17, Number 3, 2009.

Gil, Y., Kim, J., Florez, G., Ratnakar, V., and P.A. Gonzalez-Calero. "Workflow Matching Using Semantic Metadata." Proceedings of the Fifth International Conference on Knowledge Capture, Redondo Beach, CA, September, 2009.

Gil, Y., Groth, P., Ratnakar, V., and C. Fritz. "Expressive Reusable Workflow Templates." Proceedings of the Fifth IEEE International Conference on e-Science, Oxford, UK, December 9-11, 2009.

Gil, Y., Ratnakar, V., Kim, J., Gonzalez-Calero, P. A., Groth, P., Moody, J., and E. Deelman. "Wings: Intelligent Workflow-Based Design of Computational Experiments." To appear in IEEE Intelligent Systems, 2010.

Gil, Y., Gonzalez-Calero, P. A., Kim, J., Moody, J., and V. Ratnakar. "A Semantic Framework for Automatic Generation of Computational Workflows Using Distributed Data and Component Catalogs." To appear in the Journal of Experimental and Theoretical Artificial Intelligence, 2010.

Kim, J., Ewa Deelman, Yolanda Gil, Gaurang Mehta, Varun Ratnakar. "Provenance Trails in the Wings/Pegasus Workflow System," Concurrency and Computation: Practice and Experience, 20(5), 2008.

Ma, J., Shaw, E., and J. Kim. "Computational Workflows for Assessing Student Learning". Proceedings of the International Conference on Intelligent Tutoring Systems (ITS), 2010.

Kumar, V., K. Tahsin, Ratnakar, Kim, J., Mehta, Vahi, K., Nelson, Y., Sadayappan, P., Deelman, E., Gil, Y., Hall, M., and J. Saltz. "Parameterized Specification, Configuration, and Execution of Data-Intensive Scientific Workflows." Cluster Computing Journal, 13, 2010.

Reich, M., Liefeld, T., et al. "GenePattern 2.0". Nature Genetics 38(5):500-501, 2006.

Taylor, I., Deelman, E., Gannon, D., Shields, M., (Eds). Workflows for e-Science, Springer Verlag, 2007.