Workflow Evolution: Tracing Workflows Through Time

Eran Chinthaka[1,2,3], Roger Barga[1,4], Beth Plale[2,5], and Nelson Araujo[1,6]

[1]Microsoft Research, Redmond, Washington
[2]School of Informatics and Computing, Indiana University, Bloomington, Indiana
·{[3]$echintha$,[5] $plale$}@$cs.indiana.edu$, {[4]$barga$,[6] $nelsona$}@$microsoft.com$

Abstract

Scientists working on eScience environments use workflows to carry out their experiments. Since the workflows evolve as the research itself evolves, these workflows are a good tool to track the evolution of the research. Scientists can trace their research and associated results through time or even go back in time to a previous stage and fork to a new branch of research. In this paper we introduce the workflow evolution framework (EVF), which is implemented in the Trident workflow workbench[5]. The primary contributions of the EVF include i) management of knowledge associated with workflow evolution and ii) enabling reproducible research. Since we believe evolution can be used for workflow attribution, our framework will encourage researchers to share their workflows and get the credit for their contributions.

1. Introduction

Computational science experiments often involve a sequence of activities to be carried out, with a set of configurable parameters and input data, producing outputs which will be analyzed and evaluated further. Depending on these outputs, scientists will tweak input parameters, input data, and activities of the experiments and even the flow of the experiment, to improve experiment results. If the activities of the experiment or parts of the experiment can be automated, scientists will create workflows to carry out their experiments repeatedly in an efficient manner. Especially if those experiments are dependent upon the analysis of massive data sets or demand large computation resources, scientific workflows are a better option to use within them. In the workflow scenario, rather than doing everything manually, a scientist will encode their algorithms and experimental procedures as workflows and use the flexibility, tools and features of scientific workflows. When a workflow framework is used over an extended duration, the research will likely evolve along different dimensions affecting and evolving the associated worklfows(s) as well. After a period of time these scientists may need to review what they have done for a variety of reasons, possibly going back in time weeks or months. Even in operational settings, where workflows are used to produce daily results such as data cleaning and loading, these operational workflows will periodically change. We have identified through discussions with users of workflow systems several reasons why researchers may want to follow the evolution of workflows:

- They might want to see the evolution of their research. For example, if they have a better algorithm at this point, they might want to know the path it took to come to the current state and what the previous versions were.
- They might even want to go back to a previous stage. May be they want to take their research now in to a different direction and they see the best place to do that is to take the research as it was 6 months back and fork from that point.
- Sometimes scientists might discover errors in their algorithms or the experiment and want to trace back to the origin of that error. Or they might want to see the data products and results affected by this error.
- Scientists might want to visualize the data products their experiments produced over the time and use them for various evaluation purposes.

In addition to tracing the workflows over time, scientists may also be interested in re-producing workflows. In this research we introduce Workflow Evolution Framework (EVF) to help scientists manage knowledge encoded in their

workflow executions and also enable them to reproduce research.

This paper is organized as follows. In Section 2 we discuss previous efforts related to our work. Section 3 explains how workflow evolution can be used to manage the knowledge associated with workflows. In Section 4 reinforces the point of the importance of re-producible research. Section 5 explains the paper's contributions in detail. Section 6 gives a brief overview of the Trident Scientific Workflow Workbench, where our workflow evolution framework is based. Section 7 explains the approach we take to achieving our objective. Section 8 explains our application of the concepts of EVF framework in a real world scenario to track the evolution of research embedded in Word documents. Section 9 conclude the paper.

2. Related Work

There are numerous tools [16], [4], [9], [10] [5] available today for scientists to compose and orchestrate workflows. Most of these tools provided different feature sets to be used with experiments and selection of a given tool depends on the usability of a tool in a user's domain. Workflows have become so useful that they have become part of almost every major eScience platform [11], [21]. These workflows define the machinery for coordinating execution of scientific services and linking together the resources involved in an experiment. Workflows also help the scientists encode repetitive and mundane work enabling them to focus on the science. Once created, most workflows can be shared with others which helps establish best practices, but also improves the productivity of the entire research community. Researchers use provenance frameworks to gather derivation history of data products[20] [12]. Scientists argue that the provenance of data can be used to deduce the workflow traces. Since most workflow frameworks require explicit invocation of provenance handlers to capture provenance, scientists argue for the automatic generation of provenance data by workflow enactment engines that can managed provenance through underlying storage services[6]. Researchers are interested on lineage information because this information is important to properly document the scientific experiments[7]. The Earth System Science Workbench[14] uses lineage information to detect errors and determining the quality of the data sets. CMCS[17] system used lineage information to establish the pedigree of the datasets they were using. Workflow evolution has been studied previously. Vistrails[13] for example provides functionality to capture and track workflow evolution. The tool also provides a workflow orchestration environment for visualization experts to compose workflows. The system underneath captures the information required to produce the evolution of workflows. EVF, on the other hand, covers much broader aspects of workflow evolution as follows:

- Since the underlying workflow enactment engine of Trident is widely adopted and uses the Windows Workflow Foundation[3], scientists have access to a rich set of workflow definition semantics.
- Also we believe workflow evolution can also be used for attribution of work thereby people are motivated to share and use others work, in addition to the technical aspects of it.

Casati[8] introduces the dynamic aspects of workflow evolution within a workflow engine. The paper talks about the complexities of evolving workflows when under the condition of running instances. We examine our problem in a slightly more abstract sense and limit our examination to static workflows. We define workflow evolution over an extended time duration and are not limited to the runtime of an average workflow.

Scientists also have also looked at versioning of files, using different methods [19] [18]. In our framework we enable the storing of meta-data about all the versioned artifacts, but rely on the underlying data storage system to retrieve the versioned data.

3 Enabling Knowledge Management Through Workflow Evolution

Workflows encapsulate vast amount of knowledge associated with scientific experiments. We believe tracking the evolution of workflows will help to aggregate this knowledge and analyze them later for various different reasons.

3.1 Tracking Effects Over Time

When scientists associate their research with workflows, tracking the evolution of these workflows will become a very good approximation to the initial problem of tracking the evolution of their research. Along the evolution of a workflow, all the components within it will also evolve. Scientists should be able to look at the result of a workflow execution and be able to reason how the research came to current level to produce that particular output. Another important aspect of tracking the evolution is to track the lineage and the roots of errors in the experiment. For example, if an error is found in an algorithm or an input to the workflow, the evolution information should be helpful for the scientists to track back in time to find the root of the error or the affected experiments due to these errors. This information can be useful to revert back to the last known good configuration and then start research from that point onwards. Lineage information is also useful to determine the quality of the data products or the results of an experiment.

3.2 Comparing Results

A given research might evolve in more than one direction. It is really important to understand the changes on these directions by comparing the difference between the outputs of two or more versions of the same research. For example, given two outputs of two different versions of a workflow, one should be able to deduce the reason for the difference between the two results, by looking at the lineage information.

3.3 Attribution

Once a workflow is executed, it is very important to record the information like who performed the experiment, who owns or created the workflow, who owns the data products, etc., This attribution information will later be useful to track down the issues or to give proper credit to the original owners. Also, while carrying out experiments it is often a best practice to capitalize on the experiments that are already carried on the same area of research. There may be research already conducted that can be reused in the current experiment. Scientists can utilize not only the previous algorithms and implementations, but also the data products generated including optimally derived model parameter configurations. For example, in a Natural Language Processing application, scientists can use already existing bilingual corpora published by standard bodies to test their systems. Obviously these information will not only cut down the time to be spend if a scientists has to work on, but also will increase the acceptance of the current research. In research, it's not only the technical aspects that will matter. We also think sharing and attribution of research is an integral part of research. When we improve our research, we also will consider the contributions we get from other research projects as well. For example, we can extract existing modular parts from places like myexperiment.org[15] and improve our research. If we can track these contributions also within our evolution framework, this will not only give us a way to find out the contributions we had, but also will provide a way for attribution of proper credibility to the contributors, which we believe is an important aspects of scientific research from social point of view. At the same time, scientists are motivated to publish their work and like to see their work being appreciated and attributed properly. We believe a workflow evolution framework should also support work attribution. If a workflow uses work from other research, current workflow should have a way of attributing to previous work.

3.4 Provenance Information Collection

The information model of workflow evolution overlaps, to a significant extent, with workflow provenance information. Because of this, information collected using the workflow evolution framework can also be used as the basis for deducing provenance information.

4 Reproducible Research

Scientists run workflows and collect enough information to be included in the research papers they are publishing or to proceed with the current research direction. But after some time, these scientists or the other interested parties may want to revisit experiments for a number of reasons including:

- Scientists publishing research work might want to let the readers of their work also to run their experiment again to prove the results
- After considerable amount of time, scientists might need to change few parameters, inputs or configurations and re-run an experiment initially ran long time back
- Reproducibility will also increase the confidence of published research, because anyone can run the experiments and verify the results.
- Published results together with method of experiments is very important in bio-medical research to prevent adverse effect on public. Reproducibility eases this burden because more people being able to run the experiment make sure the errors are discovered faster.

For a research work to be reproduced, in the case of workflows, proper versioning of the workflow and the associated data products, parameters, configurations and executable should exist, and be bound together. The Trident Workflow Evolution framework enables reproducible research by persisting all the information about already executed experiments. If the underlying data management services enables accessing of versioned data products, then using EVF a scientist can re-run previously executed experiments. Section 8 describes a practical implementation of reproducible research we implemented using EVF by embedding workflows inside Word documents.

5 Contributions

Our contributions to workflow evolution for the management of the knowledge associated with workflow executions and management are several, and are provided in more detail below.

5.1 Unique Association of Research Artifacts to Workflows

Once a workflow is executed, it is very important to associate the associated data, parameters, configuration information and also the meta-data containing the information like who performed the experiment, when and where the output was saved. In addition to these information related to the current instance of the workflow, we also need to keep track of the lineage of the workflow itself. These unique associations will not only help to manage the knowledge associated with the workflow, but also will help to re-produce the same research at a later time.

Within the EVF framework, we enable these unique associations by recording this information inside our information model.

5.1.1 Information Model

Figure 1 shows the data structure capable of recording all the information required to track workflow evolution.

- Each workflow a user creates and the execution of that workflow is recorded inside the system, together with the associated meta-data containing information like who own/ran the experiment, the time frames, validity period, etc.,
- Each workflow execution is associated with the workflow template used to run the experiment.
- All the data products used and generated in a workflow execution is associated with the corresponding workflow instance, enabling to track them back later.
- Each workflow has links to the direct evolution (unless it's the first workflow in the evolution), which will point to the next version of the workflow, if any, and to the contributions. These contributions track the previous work this workflow is using inside it attributing to the previous work.

All this information will be persisted within a registry, in our case Trident Registry. A new version of a workflow will be saved, creating the next version when the user explicitly decides to save the workflow. But information about workflow instances and data products will be saved automatically. This information model also supports the attribution of work using the two dimensions of workflow evolution.

5.1.2 Dimensions of Workflow Evolution

We conjecture that evolution will happen along two orthogonal paths, namely *direct evolution* and *contributions*. Direct evolution happens when a user of the workflow perform one of the following actions.

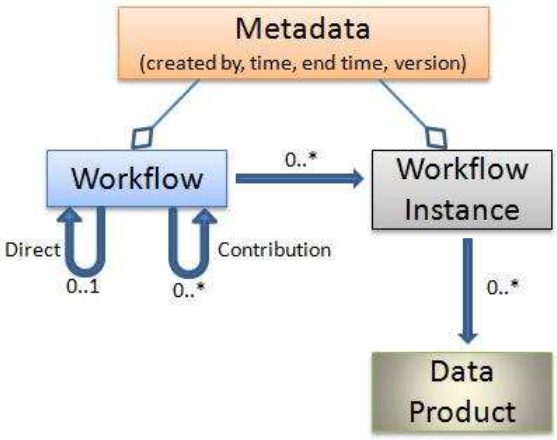

Figure 1. Information Model within EVF

1. Changes the flow and arrangements of the components within the system
2. Changes the components within the workflow
3. Changes inputs and/or output parameters or configuration parameters to different components within the workflow

For example, a scientists might change the implementation details of an algorithms used within the system, and adds that components to the workflow, removing the previous one. "Direct evolutions" will mainly come from the scientists directly involved in the research we are tracking. On the other hand "Contributions" will track the components we re-use from previous system. For example, a scientist will extract a BLAST processing module from an existing workflow, available on the Web and use it within his experiments. Or he might create a new branch from the current research and takes the research to a new direction. In this case, the research he was doing so far will be a contribution to the new branch created. One of the unique features in our system is that we track both direct evolutions and contributions to a research. This will create a better eco-system acknowledging each others contributions to the existing research and also encourages scientists to share and use existing work.

5.2 Automatic Versioning

Trident EVF helps scientists to version workflows as and when they edit them. This enables a researcher to later retrieve a previous workflow for viewing or to create new branches from the previous workflows to take them in a new direction. Versioning of the workflow template inside EVF is comparable to version control system but EVF handles

data products in a different manner. EVF framework can work with different versioning systems to support different versions of the data products and the framework provides clearly defined extension points to add new versioning systems. Within the Trident registry, a user can save enough information to retrieve a given version of a data product. When EVF is associating data products with workflow executions, it will also include this versioning information, so that the correct version of the data product can be retrieved later, in case the scientist is interested in reproducing the research. Also if an extension is registered to handle the versioning system, EVF will use that extension to automatically retrieve the data and to execute the workflow within Trident. During workflow authoring process, scientist will keep on changing a workflow and might also save all the intermediate steps. But at the end he might only execute the last version of the workflow. Should the scientist opt to delete the previous versions, EVF gives the control to the user to select the versions to persist inside the registry or to remove from the system. This will not only reduce the clutter in the scientist's workspace, but also optimizes the workflow lineage information persistence.

5.2.1 Validity of a Workflow

Versioning of workflow brings up questions about the validity of a workflow. When a new version of the workflow is created, it is a questionable whether to leave the previous versions of the workflow to be still executable or not. Within EVF we leave this decision up to scientist and select one of the solutions. In our framework each workflow is assigned a unique ID and a version number VN and a time interval $[V^b, V^e)$ that represents the time interval during which a workflow was valid. A new version of workflow ID at time t is assigned a unique version number VN, and validity interval $[V^t, \infty]$. If only one version of a workflow is allowed to be active at any point in time, which is a configuration option, the previous version VN is assigned a validity interval of $[V^0, V^t)$. Validity of a workflow only restricts whether it can be re-executed or not. All the information of previous workflows can still be tracked irrespective of this option.

5.3 Navigation Through Time

Navigating through time can give a researcher a unique view on the evolution of their research. A scientist might see change or improvement in the results of their experiments over time. They may witness the effects of the different data sets being used, or may use visual evolution to determine ownership of a piece of work or to see the contribution this particular piece of research has from the previous or related work. With the information model (Figure 1) we propose in our work, navigation through time becomes easier because it captures all the information needed for the scientists to visualize the evolution of their work. Since this information model associates the workflow instances of a given version of the workflow, scientists can also see the runtime information of each and every workflow execution. We believe that providing this information along a time-line will give users more insight into their research.

6. Trident Research Platform

In designing Trident, the was to leverage existing functionality of a commercial workflow management system to the extent possible and focus the development efforts only on functionality required to support scientific workflows. The result is a smaller code base to maintain going forward, improving sustainability and manageability of the project, and an improved understanding of requirements unique to scientific workflow.

Trident is implemented on top of Windows Workflow (WF) [3], a workflow enactment engine included at no additional cost in the Windows operating system. The Windows WF extensible development model enables the creation of domain specific activities which can then be used to compose workflows that are useful and understandable by domain scientists.

The key elements of the Trident architecture, include a visual composer and library that enable scientists to visually author a workflow using a catalog of existing activities and complete workflows. The Trident registry serves as a catalog of known data sets, services, workflows and activities, and compute resources, as well as maintaining state for all active workflows. An execution engine supports launching workflows remotely and according to a schedule. Administration tools are provided to allow users to register and manage computational resources, publish workflows for external use, and track all workflows currently running or recently completed. Users can also schedule and queue workflow execution based on time, resource availability, etc. A set of community tools includes a web service that enables users to launch workflows from any web browser and a repository that facilitates the publishing and sharing of workflows and workflow results with other scientists which integrates with myExperiment.org[15]. At the lowest level of Trident is a data access layer that abstracts the actual storage service that is in use from the running workflows. The data access layer is extensible and currently Trident supports a default XML store and SQL Server for local storage, and Amazon S3 [1] and SQL Server Data Services (SSDS) [2] for cloud storage. WF provides several runtime services which can be used as required by attaching the service implementation to the workflow runtime. Two of the most useful for our implementation of Trident are:

- Tracking service: This service enables event based tracking of a running workflow through the use of extensible tracking profiles

- Persistence service: This service allows the workflow executor to serialize and restore the entire working state of an in-progress workflow, allowing the executor to pause and resume workflows and archive intermediate state to any capable storage device.

7. Approach

We implemented the Trident Workflow Evolution Framework (EVF) within the Trident Workflow Workbench[5] to demonstrate practicality of achieving all the goals we discussed above. And while implemented in Trident, our evolution framework is general enough to be used in any workflow management system. We provide a convenient framework, within Microsoft Trident workflow orchestration environment, so that scientists can track the evolution of their research. They can go back and forward in time, and visualize the experiments. The can also find out the various contributions they have used during various stages of their research and also the various branches they had. Together with workflow evolution framework, Microsoft Trident Workflow environment will provide scientists a total eco-system for scientists to carry out their research. In this section we will explain how we implemented the concepts of our EVF framework inside Trident Workflow Workbench[5]. Scientists will use Trident Workflow Composition and Execution environments to create, edit and execute workflows. EVF is integrated in to both Trident composition environments and to the service registry. Once the user is done creating a new workflow or modify an existing workflow, he is required to save it inside the service registry. User can also retrieve an existing workflow from the registry. These retrieved workflows can be created by him or he can download existing work from Web resources like myexperiment.com.

Once the scientist has the workflow, he will then execute it within Trident workflow execution environment. All the data products and execution variables will be tracked and automatically stored within service registry. Evolution of workflows will be recorded and can be viewed from the service registry. Once the user goes in to the service registry, he will see all the workflows categorized by the name of the workflow. Figure 2 shows the view of Trident Registry, showing the evolution of an Oceanography workflow. There are two versions created so far and also research is progressing along a different direction with "Ocean Branch" workflow.

To check the workflow evolution, will select the intended workflow and asks for workflow evolution. Figure 3 shows

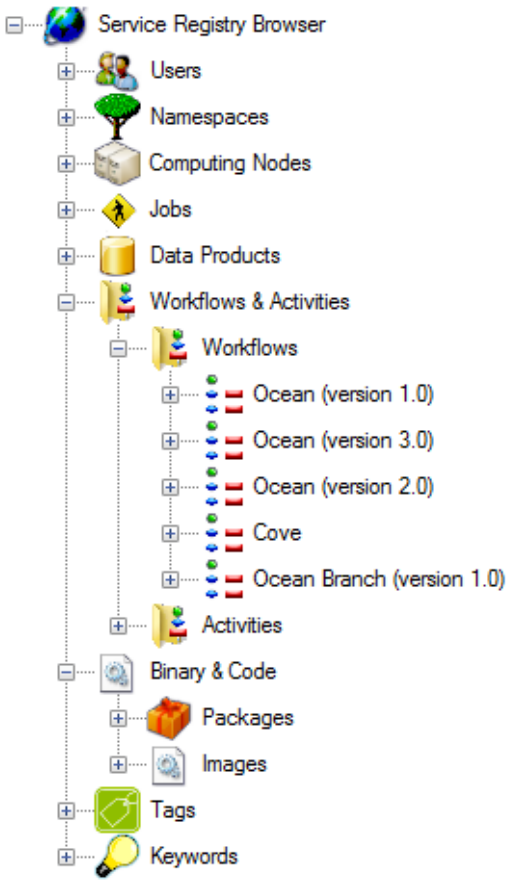

Figure 2. The Object View within Service Registry showing workflow version

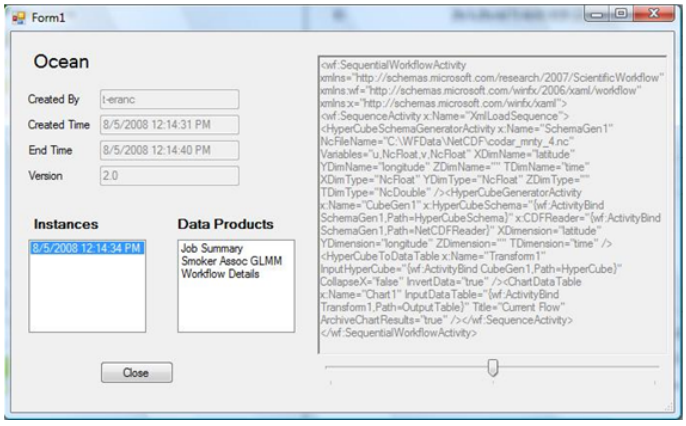

Figure 3. The workflow evolution view

the timeline view implemented within EVF. Our framework will then provide,

1. A time line view of the workflow changes

2. Meta data for each and every version of the workflow, containing information on who created, the time duration it was active and the version

3. For each of the version, user will see the workflow instances created using it and the data products consumed and generated within them. User will have the options of visualizing the data products generated within them

4. Related contributions associated for each and every workflow and the direct evolution information

Time-line view (shown in Figure 3) enables navigation, through time, by moving the time slider back and forward. This view also enables see all results that a particular workflow version created, along with the ability to select a result and track back to the workflow version that created it.

8 Applications

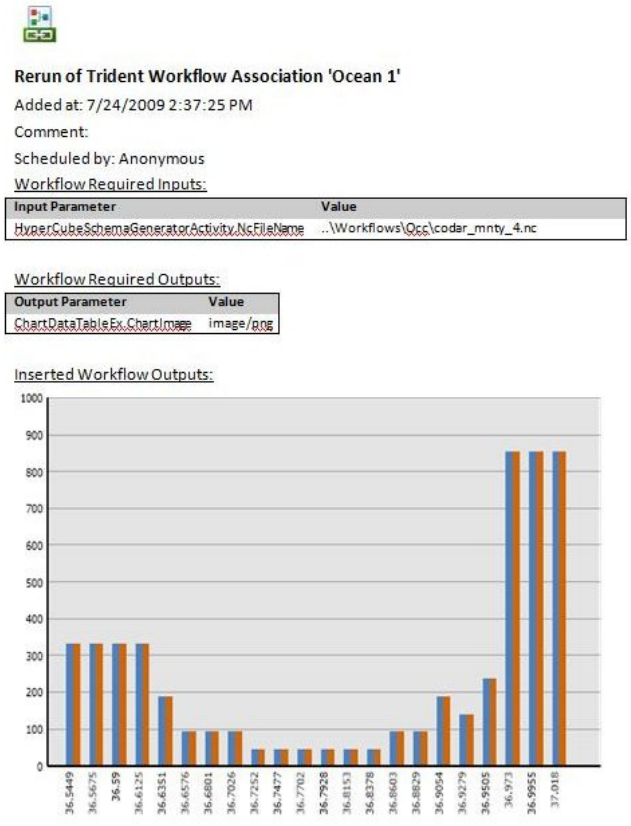

Figure 4. Embedded Reproducible Workflow with Output

Research papers include visualizations of results. But a problem with static visualizations experimental results is that it is often difficult to reproduce these results at a later time, either by the authors themselves or by the readers, for reasons that are several:

- The artifacts associated with an experiment, including metadata about the versions and locations of inputs, parameters, workflows, etc. are not recorded or tracked,
- Even if the information is available, collecting the data and getting it into a runnable state is difficult for a third party who wants to reproduce results, and finally
- If there is a change to parameters or any other artifact, scientists will have to manually run the experiments, copy the results and re-insert into the paper.

The Microsoft Trident project contains a Word plug-in which help the scientists embed experiments inside research documents. Once included, scientists can insert the outputs of these workflows, like visualizations of the results, into the Word document (Figure 4) At a later time, both by the reader or by the author itself, can run these embedded workflow to re-produce their research and to re-produce these results. This plug-in let a user to rerun the embedded workflows on a different Trident server. The add-in can be used to organize jobs, results, and workflow from various Trident servers, helping you manage research more easily. The Trident Word plug-in uses the capabilities of EVF framework to make this functionality possible, enabling the scientists to re-produce their research.

9. Conclusion

Trident Workflow Evolution Framework (EVF) allows scientists to effectively manage the knowledge around the workflow authoring and execution by tracking the workflow evolution and also enables reproducible research. The Project Trident contains an implementation of EVF framework to prove the practicality of the framework, but the concepts can be implemented in any enactment engine.

References

[1] *Amazon S3 Web Service*. http://aws.amazon.com/s3.

[2] *Microsoft SQL Server Data Services (SSDS)*. www.microsoft.com/sql/dataservices/default.mspx.

[3] *Windows Workflow Foundation*. http://msdn.microsoft.com/en-us/netframework/aa663328.aspx.

[4] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and

execution of scientific workflows. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423–424, 2004.

[5] R. Barga, J. Jackson, N. Araujo, D. Guo, N. Gautam, and Y. Simmhan. The trident scientific workflow workbench. *eScience, IEEE International Conference on*, 0:317–318, 2008.

[6] R. S. Barga and L. A. Digiampietri. Automatic generation of workflow provenance. In L. Moreau and I. T. Foster, editors, *Proc. of the International Provenance and Annotation Workshop*, volume 4145 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 2006.

[7] R. Bose and J. Frew. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.*, 37(1):1–28, 2005.

[8] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Workflow evolution. In *International Conference on Conceptual Modeling / the Entity Relationship Approach*, pages 438–455, 1996.

[9] D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor, and I. Wang. Programming scientific and distributed workflow with triana services. *Concurrency and Computation: Practice and Experience*, 18(10):1021–1037, 2006.

[10] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, and M. Livny. Pegasus: Mapping scientific workflows onto the grid. *Grid Computing*, pages 11–20, 2004.

[11] K. Droegemeier, V. Chandrasekar, R. Clark, D. Gannon, S. Graves, E. Joseph, M. Ramamurthy, R. Wilhelmson, K. Brewster, B. Domenico, et al. Linked environments for atmospheric discovery (LEAD): A cyberinfrastructure for mesoscale meteorology research and education. *20th Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, 2004.

[12] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao. Chimera: Avirtual data system for representing, querying, and automating data derivation. In *SSDBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, pages 37–46, Washington, DC, USA, 2002. IEEE Computer Society.

[13] J. Freire, C. T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo. Managing rapidly-evolving scientific workflows. In L. Moreau and I. T. Foster, editors, *IPAW*, volume 4145 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 2006.

[14] J. Frew and R. Bose. Earth system science workbench: A data management infrastructure for earth science products. In *SSDBM '01: Proceedings of the 13th International Conference on Scientific and Statistical Database Management*, page 180, Washington, DC, USA, 2001. IEEE Computer Society.

[15] C. A. Goble and D. C. De Roure. myexperiment: social networking for workflow-using e-scientists. In *WORKS '07: Proceedings of the 2nd workshop on Workflows in support of large-scale science*, pages 1–2, New York, NY, USA, 2007. ACM.

[16] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Res*, 34(Web Server issue), July 2006.

[17] C. Pancerella, J. Hewson, W. Koegler, D. Leahy, M. Lee, L. Rahn, C. Yang, J. D. Myers, B. Didier, R. McCoy, K. Schuchardt, E. Stephan, T. Windus, K. Amin, S. Bittner, C. Lansing, M. Minkoff, S. Nijsure, G. von Laszewski, R. Pinzon, B. Ruscic, A. Wagner, B. Wang, W. Pitz, Y.-L. Ho, D. Montoya, L. Xu, T. C. Allison, W. H. Green, Jr., and M. Frenklach. Metadata in the collaboratory for multi-scale chemical science. In *DCMI '03: Proceedings of the 2003 international conference on Dublin Core and metadata applications*, pages 1–9. Dublin Core Metadata Initiative, 2003.

[18] M. J. Rochkind. The source code control system. *IEEE Trans. Software Eng.*, 1(4):364–370, 1975.

[19] D. J. Santry, M. J. Feeley, N. C. Hutchinson, and A. C. Veitch. Elephant: The file system that never forgets. In *Workshop on Hot Topics in Operating Systems*, pages 2–7, 1999.

[20] Y. L. Simmhan, B. Plale, and D. Gannon. Karma2: Provenance management for data-driven workflows. *Int. J. Web Service Res.*, 5(2):1–22, 2008.

[21] P. Watson. e-science in the cloud with carmen. *Parallel and Distributed Computing Applications and Technologies, International Conference on*, 0:5, 2007.