Iterative Workflows for Numerical Simulations in Subsurface Sciences

Jared Chase, Karen Schuchardt, George Chin, Jr., Jeff Daily, and Timothy Scheibe
Pacific Northwest National Laboratory
{jared.chase, karen.schuchardt, george.chin, jeff.daily, timothy.scheibe}@pnl.gov

Abstract

Numerical simulators are frequently used to assess future risks, support remediation and monitoring program decisions, and assist in design of specific remedial actions with respect to groundwater contaminants. Due to the complexity of the subsurface environment and uncertainty in the models, many alternative simulations must be performed, each producing data that is typically post-processed and analyzed before deciding on the next set of simulations. Though parts of the process are readily amenable to automation through scientific workflow tools, the larger "research workflow" is not supported by current tools. We present a detailed use case for subsurface modeling, describe the use case in terms of workflow structure, briefly summarize a prototype that seeks to facilitate the overall modeling process, and discuss the many challenges for building such a comprehensive environment.

1. Introduction

Like other sophisticated scientific analysis problems, computational subsurface modeling presents significant complexity in preparing and executing simulations, analyzing results, and tracking data derivation. The complexity increases as larger problems are considered. For example, to complete a field-scale study, a scientist must acquire and apply sparse data samples, run simulations over extensive time scales (requiring supercomputers and generating potentially terabytes of data), and apply data reduction techniques and/or parallel visualization tools to analyze the data. The overall process is not readily amenable to automated workflow since each step usually involves human-centered analysis and decision making. However, parts of the process can benefit significantly from automation.

As described by Wainer [1], scientific workflow has a couple of important characteristics that distinguish it from business workflow. First, it is primarily concerned with generating, collecting, and analyzing large amounts of heterogeneous data. Second, and importantly, the workflow is often not completely defined before it is begun. The scientist performs some tasks and decides on further steps only after evaluating the previous ones. The implication is that a workflow is a series of "partial workflows." We call the process of data preparation, execution, analysis, and decision making followed by more data preparation, execution and analysis and so on as "research workflow," though it occurs over long time periods, requires significant user interaction, and often involves ad hoc changes as new information is discovered. It is our goal to create a user environment that supports "research workflow."

We envision an environment that provides data management for all the artifacts generated during the research process (automated or not), fully documents data derivation (provenance), and leverages workflow tools for automation of partial workflows. In other words, we seek to employ both prescribed workflow and workflow through activity monitoring. The benefits of such an environment are numerous. It can provide verifiable notebook-type records of data derivation while supporting reproducibility of results. It can enable scientists to better manage the process and understand the simulations and data weeks or months later. Through the use of automated workflow tools, it can reduce the complexity and overhead of running and monitoring simulations, and can enable the most effective use of computational resources. Finally, the provenance records could be used to support repeatable partial workflows of processes that were recorded but not prescribed ahead of time.

Recent work in scientific workflow has focused primarily on design and execution environments for prescribed "partial workflows" [2, 3] and tend to be geared toward processes with limited user decision-making. Grid workflow [4] is concerned primarily with scheduling of resources to process large, distributed data sets. We seek to leverage these types of tools but provide a more encompassing environment that supports the long running, ad hoc nature of computational research. We are building this system as a series of prototypes. Our goal for the initial prototype was to develop an architecture and demonstrate the integration of workflow automation tools, data and provenance services, analysis tools, and workflow monitoring, all organized in a meta workflow environment, while also building a deeper understanding of workflow in the context of subsurface sciences.

2. Background

At sites such as the Hanford Nuclear Reservation in southeastern Washington state, large volumes of radiologically and chemically hazardous “legacy waste” were released into the subsurface environment during the development and manufacture of nuclear weapons and nuclear reactors. These contaminants subsequently have migrated away from the disposal sites through the subsurface environment and toward potential environmental receptors such as rivers and wells. Numerical simulators are frequently used to assess future risks, to support remediation and monitoring program decisions, and to assist in design of specific remedial actions. Because of the high degree of spatial variability (heterogeneity) in the subsurface, limited access for characterization of subsurface properties, and incomplete understanding of physical, chemical and biological processes, there exists a high degree of uncertainty in model predictions. Many alternative simulations may be performed, each of which can generate large output data sets that require post processing analysis, visualization, and archival. In this paper, we explore the use of workflow techniques and technologies to enable both engineering and scientific research uses of complex models of subsurface flow and contaminant transport.

In our test case, the numerical simulations are applied to an experimental study of two solutes mixing and reacting to form a mineral precipitate. The engineering application of interest is the potential for in situ immobilization of radioactive groundwater contaminants by manipulation of groundwater chemistry to induce calcite precipitation. Future experiments are now being planned and must consider complex flow patterns, flow rates, and solute concentrations. Simulations support the design by evaluating a number of alternative scenarios. Although these simulations are relatively simple in comparison to field-scale site simulations, they represent an opportunity to define a generalized workflow process for scientific computation of reactive transport problems and to test a preliminary workflow framework.

3. A use case

Completing the calcite precipitation work involves running a series of related simulations. The following is a high level description of the actual workflow used to carry out the initial part of the study but condensed for readability. See Figure 1 for graphical representation of the process.

First, a model of the grid is developed. This model will be used throughout this example. This is primarily a manual process and is performed outside the environment. Referring to Figure 1, a set of material properties distributed over the grid is developed by running a small FORTRAN program (a). Following this, an input file is specified (b) and a steady-state “flow” simulation is run (c). The restart file from this simulation is then used to run a “transport” simulation (e) with a revised set of parameters (d). An analysis tool is used to view the results (g) after some data processing (f). We refer to this process as partial workflow 1 or “PW1.”

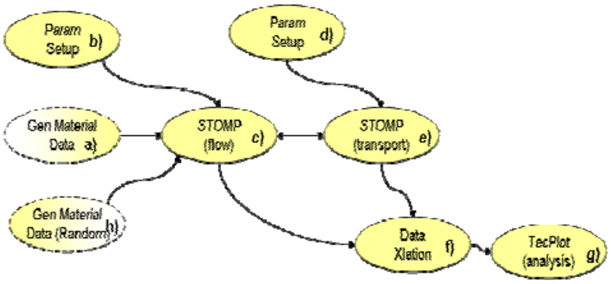


Figure 1: Partial Workflow 1 (PW1) and Variations

This partial workflow, and variations of it, is run numerous times. First, a pair of simulations is run using the same materials but with revised parameters. The new input files are derived from the input files in PW1. This partial workflow (PW2) is structured similarly to PW1 except that there is no step to generate material data type, i.e., there is no (a) in the graph. Following the user analysis at each phase, a series of partial workflows that are very similar to the first two (PW1 and PW2) are then performed. A third variation (PW3) is defined when a new method for defining material types is used (h). A graphical view of the overall “research workflow” is shown in Figure 2 where each process represents the sequence of details shown in Figure 1. This graph shows a great deal of complexity; there are branches of investigation and cross links between branches resulting from the way new simulations are derived from existing simulations. In terms of automated workflows, the primary use is to perform the tasks associated with distributed computing though it might be possible to generate post processed results.

4. Prototype design and implementation

Our system architecture consists of four major components as shown in Figure 3. The data services component tracks artifacts of the research process and integrations with archival storage for large data sets. The visualization component is simply a mechanism for integrating a variety of third party visualization tools. The workflow component is used to specify and execute automatable sequences of processes. Finally, there is an "Organizer" component that provides a user view onto past simulations, and integrates tools for data processing, simulation setup, and analysis. It is the glue that ties together the "partial workflows."

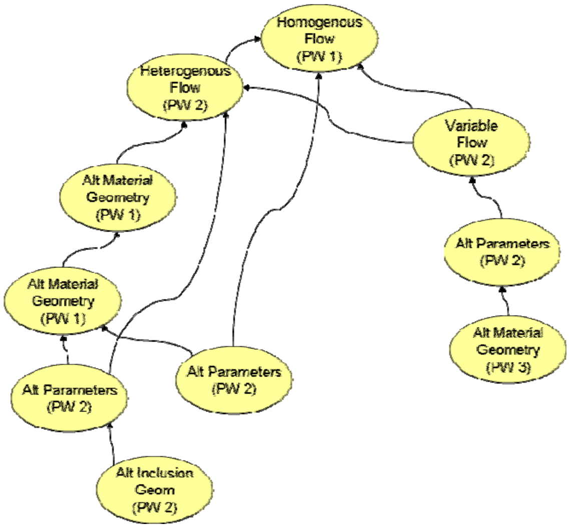


Figure 2: Overall Workflow

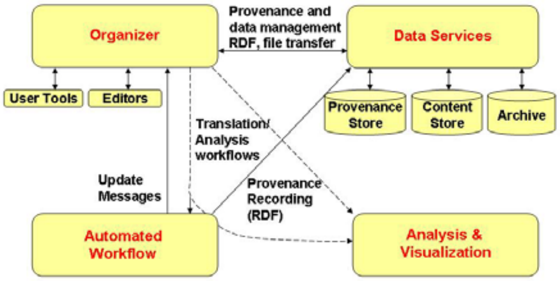
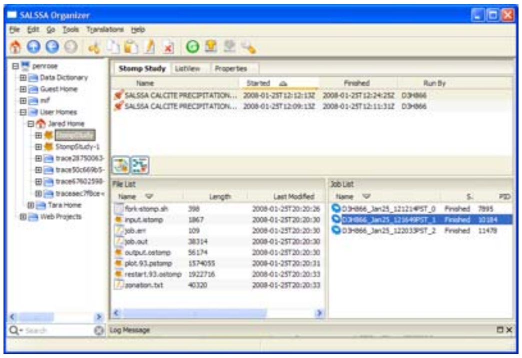


Figure 3: System Architecture

4.1. Organizer

Current workflow environments are not designed around the concept of partial workflows. Therefore, we have constructed our own tool that can manage access to shared workflows, interact with a workflow engine, integrate tools and translations that can be used outside of automated workflows, and present a view onto the user's workflow and associated data. We call this tool the "Organizer." It enables users to track their work within computational studies and derive new runs and studies from existing ones after performing interactive analysis. Figure 4 shows a picture of the Organizer with the studies and other data on the left and the details of a study shown in the panel on the right. The study view currently shows a "dashboard" type summary of runs, the jobs within each run, and the files associated with any given job.



Name	Started	Finished	Run By
SALSSA CALCITE PRECIPITATION...	2008-01-25T12:12:13Z	2008-01-25T12:24:25Z	D3H866
SALSSA CALCITE PRECIPITATION...	2008-01-25T12:09:13Z	2008-01-25T12:11:31Z	D3H866

Name	Length	Last Modified
fork-stomp.sh	398	2008-01-25T20:20:26
input.istomp	1867	2008-01-25T20:20:30
job.err	109	2008-01-25T20:20:30
job.out	38314	2008-01-25T20:20:30
output.ostomp	56174	2008-01-25T20:20:30
plot.93.pstomp	1574055	2008-01-25T20:20:31
restart.93.ostomp	1922716	2008-01-25T20:20:33
zonation.txt	40320	2008-01-25T20:20:33

Name	S.	PID
D3H866_Jan25_121214PST_0	Finished	7895
D3H866_Jan25_121649PST_1	Finished	10184
D3H866_Jan25_122033PST_2	Finished	11479

Figure 4: Organizer

4.2. Data services

Our data systems consist of an RDF store for capturing arbitrary information about processes and data as a directed graph and a content system for managing workflow artifacts such as inputs, outputs, and parameters. Recording of processes happens both as the user performs operations and as automated workflow steps execute through a provenance API. During workflow execution, we monitor system events and automatically record information about each process as it occurs. Queries are constructed to provide the views shown in Figure 4 above.

4.3. Workflow design and execution environment

The primary application of workflow automation tools is to run simulations on distributed resources. This involves: file staging, job submission and monitoring, and file recovery. The workflow is executed using Kepler [2]. Aside from basic execution, the workflow has a couple of important capabilities. First, it is capable of taking a set of job specifications and launching them consecutively, keeping a resource fully loaded without requiring manual monitoring and scheduling. This is useful when conducting parameter sweeps. Second, it is capable of distributing multiple job requests across different machines thus providing a simple form of load balancing. The number of simultaneous jobs is specified as a parameter.

The workflow is instantiated with a set of input parameters provided by a wizard developed to support code setup and a job launching tool. These parameters contain information about remote compute resources as well as the specification of inputs required for running each numerical simulation. The workflow handles all of the details of job launching and monitoring and notifies the Organizer when job status changes occur.

4.4. Visualization

Visualization is currently performed with TecPlot. It is loosely integrated through a simple but general tool and data translation mechanism that can support a number of analysis tools once data translators are developed. We anticipate the need for much more complex translations that themselves can be specified in a workflow design/execution environment. For example, data may need to be moved from archive storage to an analysis machine and reduced prior to invoking a visualization or analysis tool.

5. Analysis/challenges

The user environment we envision poses many challenges. Little attention has been devoted to research workflows as there are still many challenges to be addressed in supporting partial workflows.

In terms of the Organizer and its presentation of research workflow, we recognize the need for a more general, graph-based interface to represent the full set of steps in the process and the complex relationships between steps. We envision a graph that serves both as a way to initiate next steps in the research as well as a way to represent the iterative process that has already been performed. This will provide immediate visible indication of how processes are related while allowing the user to continue to branch off the graph and try new experiments. Unfortunately, after only a few iterations, the full view of the graph becomes unwieldy as can be seen in the full depiction of our use case in Figure 5. Note that the graph shows processes. Files and other artifacts will be shown in a coordinated file panel. To address the complexity problem, powerful user definable filters, based on arbitrary attributes, must be applied to the graph. We also imagine user control over expanding and collapsing nodes, zooming, user grouping and annotation mechanisms, and multiple layout algorithms.

During the automated portion of the workflow process, provenance is recorded by listening to the Kepler execution engines events and recording every detail. If the components within a workflow are at a sufficiently high level of abstraction, this approach would work. However, designing in Kepler currently means connecting many low level components (about 50 in our case). Recording all actions creates significant "noise" in the provenance record and makes querying both difficult and slow. To solve this problem, we are investigating mechanisms to specify the provenance of interest at design time.

In our experience, current workflow tools fall far short of supporting workflow as conceptualized by researchers. For this reason, we have started down the path of creating a higher level framework. Current workflow tools are also too complex for most users and must be used "behind the scenes." This creates additional challenges with the adaptability of automated workflows in that without careful thought towards developing generic workflows that fit common patterns, the system will be fragile.

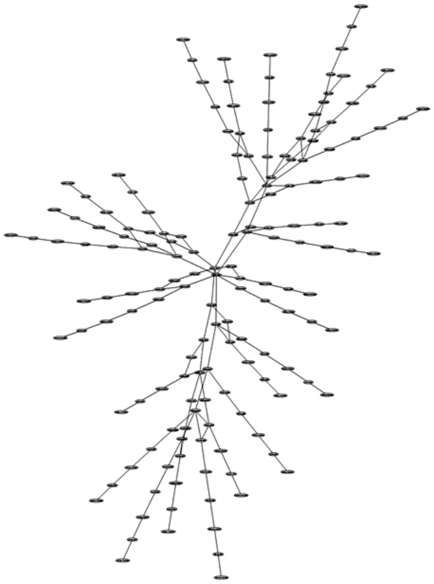


Figure 5: Full Graphical Depiction of Use Case

6. Acknowledgements

This research is supported by the U. S. Department of Energy's Office of Science under the Scientific Discovery through Advanced Computing (SciDAC) program.

7. References

[1] J. Wainer, M. Weske, G. Vossen, and C.B. Medeiros. Scientific Workflow Systems. In Proc. of the NSF Workshop on Workflow and Process Automation Information Systems, 1996.

[2] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, Yang Zhao. Scientific Workflow and the Kepler System. Concurrency and Computation: Practice and Experience. Volume 18, Issue 10, Pages 1039 – 1065.

[3] Taverna. http://taverna.sourceforge.net/. March 14, 2008.

[4] J Yu, R Buyya. A taxonomy of scientific workflow systems for grid computing. SIGMOD Record, 2005.