

Knowledge Annotations in Scientific Workflows: An Implementation in Kepler

Aída Gándara¹, George Chin Jr.², Paulo Pinheiro da Silva¹, Signe White²,
Chandrika Sivaramakrishnan², and Terence Critchlow²

¹ Cyber-ShARE,

The University of Texas at El Paso, El Paso, TX

agandara1@miners.utep.edu, paulo@utep.edu

² Pacific Northwest National Laboratory, Richland, WA

{george.chin, signe.white, chandrika.sivaramakrishnan, terence.critchlow}@
pnl.gov

Abstract. Scientific research products are the result of long-term collaborations between teams. Scientific workflows are capable of helping scientists in many ways including collecting information about how research was conducted (e.g., scientific workflow tools often collect and manage information about datasets used and data transformations). However, knowledge about *why* data was collected is rarely documented in scientific workflows. In this paper we describe a prototype system built to support the collection of scientific expertise that influences scientific analysis. Through evaluating a scientific research effort underway at the Pacific Northwest National Laboratory, we identified features that would most benefit PNNL scientists in documenting how and why they conduct their research, making this information available to the entire team. The prototype system was built by enhancing the Kepler Scientific Workflow System to create knowledge-annotated scientific workflows and to publish them as semantic annotations.

Key words: Scientific Workflows, Knowledge Annotations, Kepler

1 Introduction

When scientists work collaboratively to conduct scientific research there are many factors that have a direct impact on how research is performed, much of it implicit in the actual process used. For example, when exploring a data set, scientists may use their expertise to select data points and their experience may guide a scientist to impose a certain constraint on the entire dataset. Unfortunately, these decisions are poorly documented and may not be presented to the current research team, much less reflected in any published work. Nonetheless, this knowledge is crucial for conducting research and is the basis for innovation, something that is often needed for scientific research [13].

Scientific workflow tools enable scientists to describe, execute and preserve a research process. In addition, they can be used to annotate data and collect

provenance about how scientific artifacts were created [8]. However, the knowledge implicit in a workflow reaches beyond its execution, including, for example, the many decisions made to choose algorithms, parameters and datasets that are undocumented in current scientific workflow engines. If that information could be captured in scientific workflows, the associated tools are in a unique position to play an instrumental role in organizing and preserving the implicit and explicit knowledge that is shared among scientists during a research effort.

Similar to many active scientific research projects, the subsurface flow and transport analysis projects at the Pacific Northwest National Laboratory (PNNL), is a collaborative effort where scientists with different knowledge domains, e.g., data collection, model building, and simulation expertise, are working together to perform groundwater modeling. We evaluated the scientific process being used by the groundwater modeling team to understand how collaborative teams conduct scientific research, share knowledge, and produce scientific products. As observed with other teams affiliated with the Cyber-ShARE Center of Excellence³, we observed that scientists of a highly collaborative team are dependent on the decisions, expertise and results of their colleagues during a research effort, because assumptions made during one scientist's analysis directly affects other scientists on the team. Unfortunately, these decisions and assumptions are often not well documented and their justifications may fade over the course of the project. Providing mechanisms to help scientists manage this knowledge would be a great benefit to assure that the entire team remains aware of relevant research assumptions, constraints and decisions.

This paper explores the process of documenting the implicit side of scientific research using an extension to the Kepler Workflow Management System [10]. Although Kepler has been used to represent groundwater modeling workflows previously, the tool was not able to support documenting the implicit aspects of the research collaboration. In order to represent these decisions and assumptions, Kepler, similar to other executable workflow systems, needed to be extended. To this end, a prototype system was built over Kepler to produce knowledge-annotated scientific workflows. The collected information is published as semantic annotations in RDF [9] with the goal of enabling reuse and integration with related information [14].

In the remainder of this paper, we present a preliminary research prototype designed to address the current limitations of workflow systems and enable documentation of the entire scientific process from initial discussions to executable workflow. Section 2 will provide background information on a subsurface flow and transport project that motivated this prototype as well as technologies that affected our implementation, including Kepler. Section 3 presents the details of our implementation while Section 4 discusses current issues with this prototype and outlines future work for knowledge-annotated scientific workflows, including a comprehensive user study. Finally, Section 5 presents our concluding thoughts.

³ <http://cybershare.utep.edu>

2 Subsurface Flow and Transport Analysis Case Study

PNNL has extensive research and development capabilities and expertise in the scientific field of subsurface flow and transport, which focuses on the "study of chemical reactions in heterogeneous natural material, with an emphasis on soil and subsurface systems[1]." Some of these capabilities are centered on the construction and application of a PNNL-developed predictive subsurface flow and transport simulator known as STOMP (Subsurface Transport Over Multiple Phases)[16]. Using STOMP along with other subsurface model development software, PNNL groundwater scientists are modeling subsurface flow and transport on a wide variety of internal and external projects.

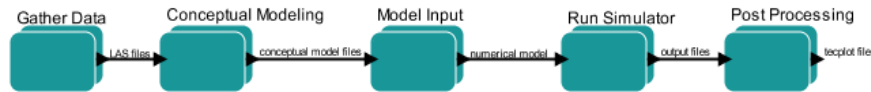


Fig. 1. A high-level workflow describing a groundwater modeling project conducted at PNNL.

A groundwater modeling project typically comprises a project manager and several team members. Scientists take on roles within the overall research effort based on their expertise. The groundwater modeling process follows a general data flow pattern of steps that have been summarized as a high-level workflow shown in Figure 1. Each step of the process requires the knowledge and expertise of the scientist performing it, as well as collaboration between steps to understand details about the overall process. The groundwater process starts with the Gather Data step, where data is initially collected by a geoscientist. This geoscientist gathers the data from the field or while running experiments. Next a scientist performs the Conceptual Modeling step where the initial data is analyzed to create a conceptual model (a conceptual understanding of the subsurface geology). The conceptual model is represented in a series of files that are used in the Model Input step. Within the Model Input step a scientist builds a numerical model of the data and annotates it. This numerical model is an implementation of the conceptual model into a discretized, numerical framework. The simulation is executed in the Run Simulator step after which the data is post-processed into target data images and reports.

Throughout the groundwater modeling process, scientists use their expertise to interpret and analyze data, interpolate new data models, run scripts and executables (some of which they have written themselves), visualize data and

results, and build and annotate data sets. They must understand details about the overall project and must make simplifying assumptions to account for a lack of data or to take into consideration computational limitations, e.g., available hardware, and project time constraints. Their research is an iterative process, where they might run a series of steps over and over, changing parameters and analysis details, to produce the best results. The collaborative team continuously reviews the results of different steps, makes suggestions, and formulates new assumptions that could alter the overall modeling process: that is, the changes to the process might require that steps be performed again. Sometimes, having to perform steps over could be avoided if each scientist was aware of assumptions or constraints that other team members have made. In many cases, scientists keep journals and notes of what worked and what did not as well as the decisions, assumptions or constraints used to find results. Team discussions normally occur in meetings, via email or by phone. Documentation of these discussions does not always occur.

We found that many of the needs of these scientists are consistent with the needs of other scientists in other domains such as geoinformatics or environmental science. Namely, once an artifact or analysis product is available either during the research process or after, scientists seek to understand not only the "hows" but the "whys" of its creation. Currently, the data and knowledge of running these models are collected in a final report and result data sets. Going back to review and understand the original process requires understanding the final report, visualizing the final results, if they are available, and talking to scientists involved in the initial research and analysis. A scientist's recollection of a specific past project would require that they too have access to their notes and the details of how they conducted scientific analysis and performed the specific steps of a specific process.

2.1 Current Approaches for Collecting the Scientists' Knowledge

One common method that scientists have used to collect their scientific notes is paper-based journals, where scientists can sketch and write the ideas and conclusions of their research. Electronic Laboratory Notebooks (ELNs) provide similar functionality to paper-based journals but with added benefits like electronically organizing scientific notes, relating notes to data, viewing and manipulating related electronic data and multi-user support. More recently, ELNs have been enhanced to function over the Web and to provide interoperable semantic annotations so they can be integrated with other data producing and consuming systems [15]. On a more collaborative front, tools like email, chat tools, and more recently social networking tools like Facebook⁴ have been used to elicit discussions and social interactions. These tools support the ability to link data to specific comments, e.g., through attachments, and to talk directly to individuals as well as groups. Although both ELNs and social networking tools promote collaboration and enable the collection of social and scientific information, they

⁴ <http://facebook.com>

are limited in documenting the ongoing research process conducted by scientists. For example, they do not directly support the process definition that is inherently captured in scientific workflow tools.

Executable scientific workflows are beneficial to scientific research and the management of scientific data [8] because their main goal is to collect sufficient information so a process can be executed. As a result, they can capture hardware details, user details, operating system details and execution details that can be used to annotate an artifact. Furthermore, the graphical representation that many scientific workflow tools create enables scientists to see the steps in the process and the flow of data. Through this representation, scientists can understand how an artifact was created. As scientific research efforts become more collaborative, scientific workflow environments must consider how they will evolve to support collaborative teams. myExperiment, for example, is a research effort where scientific workflows and other research objects can be published to the myExperiment Portal [7]. The goal of this portal is to support the sharing of scientific research and the reproduction of scientific results. Through myExperiment, users can rate, download and tag workflows as well as maintain discussions about published workflows. The documentation occurs after the workflow is published, thus knowledge annotations occur after the scientific research is completed.

Another benefit of scientific workflows is the ability they provide in reproducing results. The Kepler Scientific Workflow System, for example, publishes workflows and its corresponding data as Kepler Archive (KAR) files. KAR files are encapsulated workflows including all components needed to run the workflow [2]. As a result, KAR files are objects that can be published on myExperiment and then downloaded for other scientists to execute. WDOIT! [12] is a scientific workflow tool that publishes all workflow information as semantic annotations on the Semantic Web. With these annotations, a WDOIT! workflow can classify data with common terminology and link workflows to data and resources on the Web. This open environment enables interoperability and reuse with other semantic-based tools [14].

2.2 Kepler Scientific Workflow System

Kepler is a scientific workflow tool used to build and execute scientific workflows. Kepler provides graphical abstractions to enable scientists to build workflows. For example, scientists describe the steps of a process by adding actors to a canvas (a graphical window); they add ports for data input and output of an actor; and they connect ports between actors to specify a flow of information. Actors can be chosen from a list of available actors; there are menu entries to add ports and actor details, and workflows can further be described by specifying parameters, global values, and execution instructions. Kepler workflows can be described at multiple levels of detail using composite actors. A composite actor is a nested canvas where a scientist can define a subworkflow. The purpose of a composite actor is to hide process details until there is a need to see them, e.g., a scientist chooses to open it. Once all the details for execution have been specified

in a workflow canvas, including actors, ports, parameters, and connections, the workflow is ready for execution.

As shown earlier, Figure 1 presents a Kepler workflow representing the groundwater modeling steps identified by groundwater scientists. Building an executable workflow can be quite involved and in some cases distracting, particularly if scientists work in an ad hoc, exploratory framework where they are not certain of the steps they will take to conduct their research. Furthermore, scientists may need to describe a process but not execute it, e.g., if they use their own scripts, programs or manual steps not found in Kepler. Nevertheless, the ability within Kepler for describing processes at different levels of abstraction provides a mechanism for annotating the internal components, e.g., actors and connections, in the workflow with knowledge annotations.

3 Knowledge-Annotated Scientific Workflows

The goal of this research effort was to help the groundwater scientists manage collaborative data that is traditionally generated but not collected during a research effort. Three design principles were incorporated into this prototype. *First design principle*: the prototype needed to enable scientists to primarily describe their research; thus, any workflow construction needed to be a byproduct of the information provided by the scientist, requiring execution details only when necessary. *Second design principle*: the prototype needed to align with the way scientists conducted research and limit the duplication of information and the number of menus and windows needed to document their research. *Third design principle*: the prototype needed to leverage the workflow to manage the annotations, i.e., annotations had to directly relate to steps and connections that the scientist added to their research process, which would in turn enable the prototype to present and publish the result data with a close relation to the process. The prototype enables scientists to build workflows by focusing on scientific analysis and ad hoc scientific exploration. The following sections will work through the prototype system, highlighting the features that were added to support groundwater scientists in documenting the decisions and knowledge behind their research.

3.1 Building Workflows

To enable the groundwater scientists to focus on their research and avoid focusing on executable workflow details, the Kepler interface was modified by adding menu entries, buttons and panels to collect and display the implicit knowledge related to steps scientists take to conduct scientific research. Figure 2 shows the overall prototype interface. The left side of the interface displays the Research Hierarchy panel, which is a tree view of all the process steps defined in a project, and the bottom right displays the Process View panel which shows a graphical representation of the steps in a process. These two panels are always visible, but their content changes based on a scientist's interaction with the tool. The top

right displays various tabs that collect annotations related to specific process details, e.g., steps, inputs, outputs. Workflows in the prototype are managed

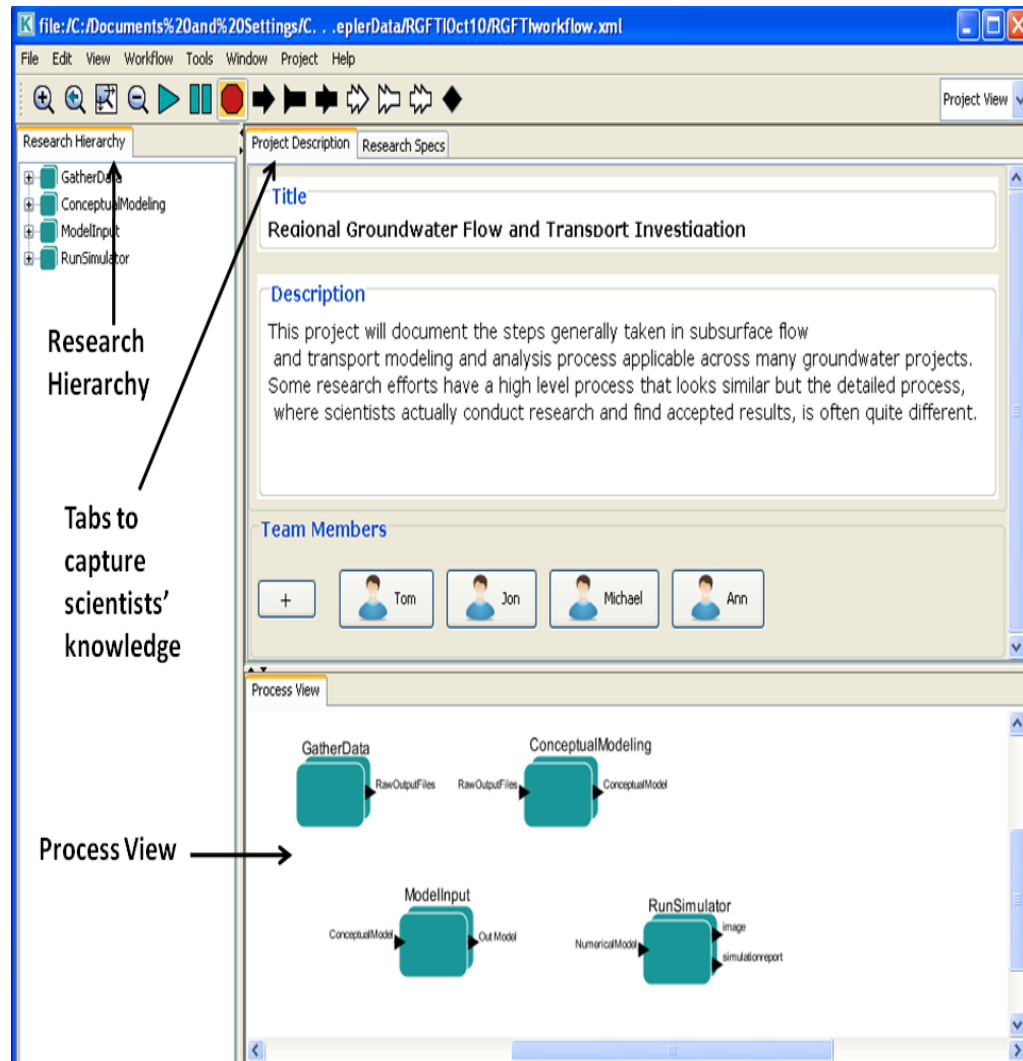


Fig. 2. The prototype interface showing the different panels added to support scientists in annotating their workflows as they describe their research.

within a project. A project is created by opening up the prototype interface and entering the project details. The Project Description tab is shown at the top of Figure 2. Here a title, description, and the project’s team members are

specified. When team members are selected for a project, a project manager enters the name and contact information for each team member. In this way, the team has a single place to identify who is actively on the team. Steps can be added to describe the analysis steps that will be performed within a process. As steps are added, a step actor is added to the Process View panel. For example, a subsurface flow and transport project would conduct some variation of the steps defined in the case study found in Section 2. The Process View panel in Figure 2 shows steps for gathering data, building the conceptual model, building the input model, and running the simulation. A scientist can add steps as needed to allow for an ad hoc, exploratory collection of scientific analysis steps. In turn, the workflow system is collecting annotations, relating them to specific components of the workflow and providing a mechanism for a team of scientists to view the information as it relates to their ongoing research.

Throughout a project, decisions and notes are made that affect the entire project. This information is collected in the Research Specs tab shown in Figure 3. The Research Specs tab has a scrollable entry pane where a project manager or other team member, can enter details about known constraints based on the research proposal, administrative comments as to how the group will function, or ongoing comments about what the group is doing. This tab also has buttons to support the annotation and workflow building process, e.g., Add Step, Add Assumption, and Add Deliverable.

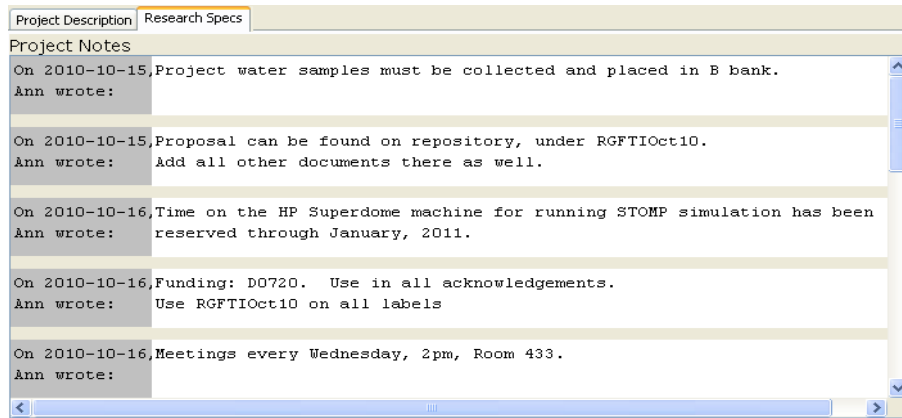


Fig. 3. The Research Specs tab. This tab collects general comments affecting the entire project and provides buttons to help scientists describe their research and annotate process components.

The prototype leverages Kepler’s ability to describe processes at multiple levels of abstraction by annotating workflow components, e.g., steps, at the different levels. The Process View panel in Figure 4 describes in more detail the

RunSimulator step conducted by a groundwater modeling scientist. At the top of Figure 4 is the Research Notes tab. When a step is created, a Research Notes tab is created with an associated workflow. The Research Notes tab has a scrollable entry pane where scientists can capture their notes about the research, e.g., the decisions they made, what processes worked and why. Scientists can also add more process details that will be reflected as changes to the workflow, e.g., Add Steps, Add Input, Add Output. Scientists can then choose to refine steps to any level of granularity. Figure 5 describes a more detailed level of granularity of the STOMP step, where the groundwater scientist models and annotates the execution of a STOMP actor, an executable actor used at PNNL. This method of defining and refining steps allows scientists to define a hierarchical definition of research where steps are refined if the scientist needs to describe more detail. More importantly, the process exhibits the integration of annotating a completely abstract process shown in Figure 2 down to a more detailed executable model shown in Figure 5, where scientists can add comments concerning the success or failure of parameters or executed processes.

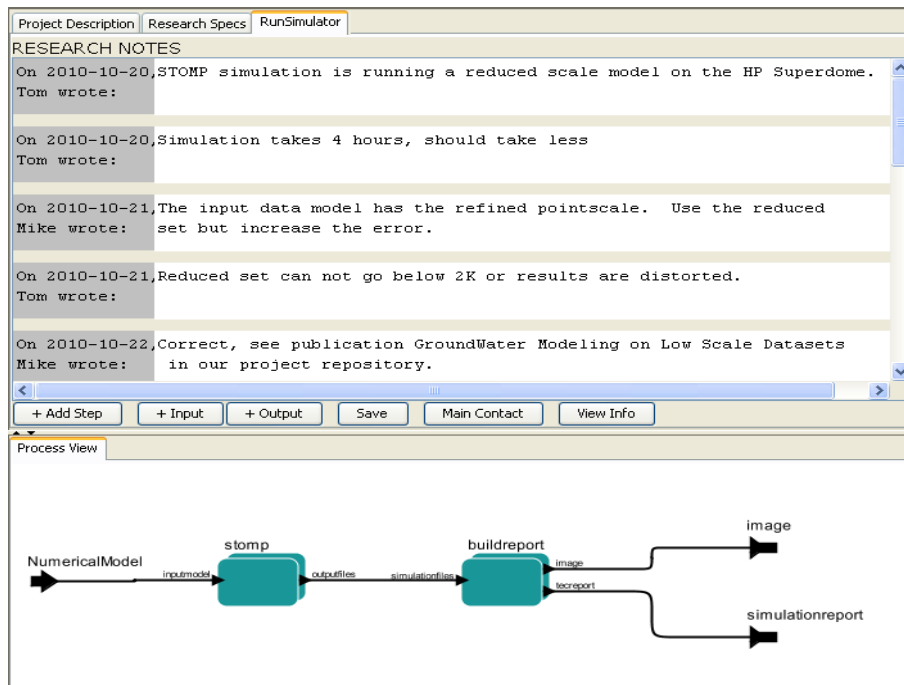


Fig. 4. The knowledge data and workflow of the Run Simulator step. Scientists can refine the details of their research by describing a process, adding research notes and entering details about the inputs and outputs of their research.

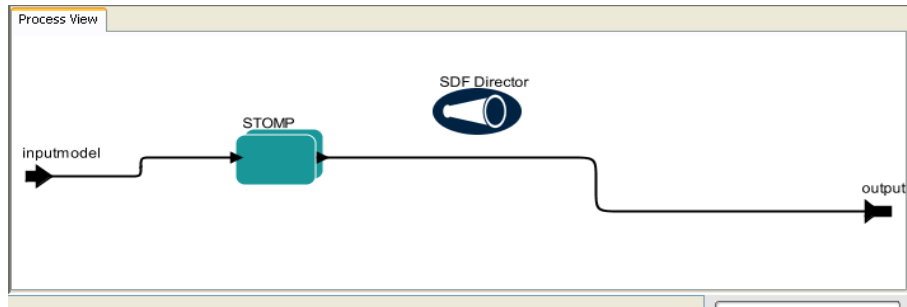


Fig. 5. The Process View panel for the STOMP step. The Process View panel has execution details for running the simulator, e.g. a director, actor, inputs and outputs.

As steps are added at any level of the process, the Research Hierarchy panel, shown in Figure 2 is updated with branches added to the appropriate hierarchy tree. When a step is opened (by double-clicking its icon in the Research Hierarchy tree or from the Process View panel), the step’s Research Notes panel and Process View panel are displayed. When other scientists need to understand the details about a particular step, they can open the step and see the annotations.

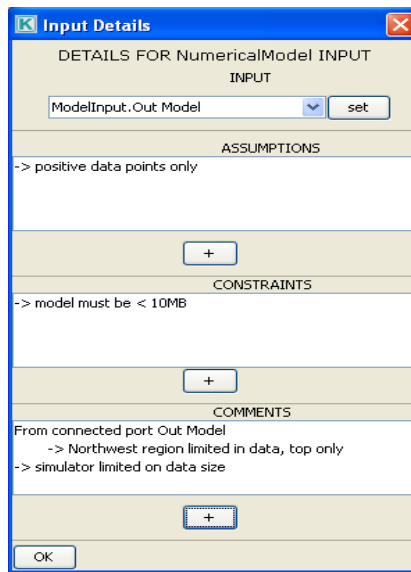


Fig. 6. Input details dialogs for the NumericalModel input of the RunSimulator step. This input is dependent on Outputfiles output of the ModelInput step. Comments from the Outputfiles output can be see here as well.

To identify data that will be used or produced in a scientific step, scientists can add inputs or outputs that are displayed as icons on the Process View panel. The Process View panel in Figure 4 shows several input and output icons, e.g., `NumericalModel` as an input and `simulationreport` as an output. A scientist can add more knowledge about a specific input or output by opening its Input Details dialog or Output Details dialog, respectively. Figure 6 shows the Input Details dialog for the `NumericalModel` input from the `RunSimulator` step. Using this dialog, scientists can specify what data is needed and from what step, specifying the flow of information and the dependencies between steps. The Output Details dialog (not shown here) allows a scientist to enter similar details related to the outputs produced within a scientific step. These dialogs collect assumptions, constraints, and general comments about the data. Collecting this information as the scientist is conducting research can provide insight to other scientists that depend on it. To make this knowledge more visible, the knowledge data of connected input and output ports are displayed in the respective Input or Output Details dialogs. For example, the comment section in Figure 6 also shows the comment from the connected step's (`ModelInput`) data. Scientists who view a details dialog can review existing details and add their own.

As a result of specifying data dependencies in the details dialogs, connections are made on the Process View panels that contain those actors, building the workflow automatically. Figure 7 shows the complete project workflow after the different steps were opened by the corresponding scientist and research knowledge was collected in them. Notice that the `NumericalModel` input to the `RunSimulator` step is connected to the `NumericalModel` output from the `ModelInput` step. This information was specified in the Input Details dialog in Figure 6.

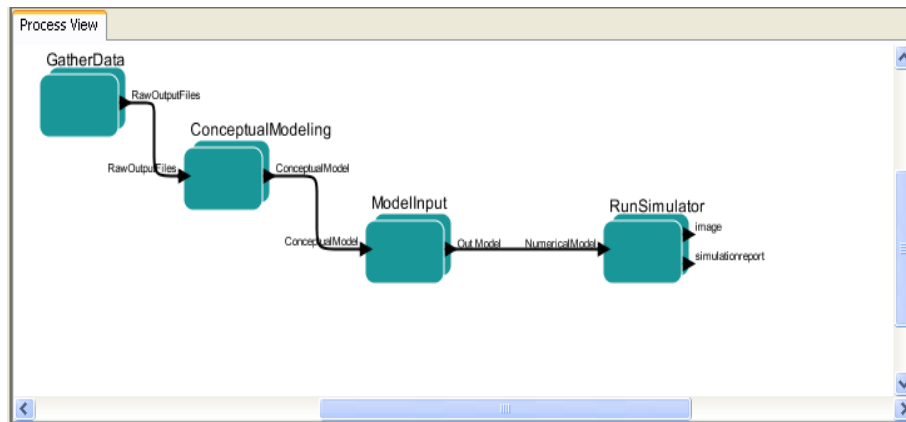


Fig. 7. This workflow was built in the prototype system by collecting knowledge about the research performed by each scientist.

3.2 A View of the Data

Having scientists enter research notes and knowledge details to describe their research is useful for data capture but not sufficient for understanding the data. For scientists to understand a step within the data collection windows provided in the prototype, they would have to open up different tabs and dialogs. By managing this information within the workflow tool, the prototype can help provide this information in a more organized format. The scientists in our case study specifically referenced a need to understand the flow of the data that was coming into their research step and the ability to summarize the details of the step they were viewing. Furthermore, the scientists are often called on to provide the same details that are collected in this prototype for a final project report. To support these needs, we added the ability for scientists to backup or move forward through step connections, build a full summary report and access summary views that can be seen at any level within the research hierarchy.

The backward and forward traversal functionality was added to enable quick jumps between steps. With a single click, a scientist can choose to step back or move forward from a port to the step that it is connected to, which displays the connected step's Research Notes and Process View panels. For example, performing a backward traversal from the NumericalModel input for the Run-Simulator step would open up the Research Notes and display the workflow for the ModelInput step. The advantage to this feature is that it makes it easier for scientists to see the research notes and process view of related steps and then jump back if needed.

To help scientists understand the entire research effort, a full research summary report can be created. The details of scientific research can be quite involved and the goal of this prototype system is to help collaborative groups document and understand the process behind their research. Building a full summary is useful because it summarizes all the details of the scientific process in one document. Another benefit of this feature is that it could facilitate writing documentation, e.g. a research paper or publication, about the scientific process and the accompanying knowledge data. Figure 8 shows the first page of a summary report for the Regional Groundwater Flow and Transport Investigation (RGFTI) project, which is a specific groundwater modeling project currently being performed at PNNL. Understanding what should be in this summary and how it should look is dependent on the group working with the data as well as the needs of each scientist. Further evaluation should help with understanding the appropriate configurations and structure of such a summary.

To help scientists understand the factors that are contributing to the current state of a step, a scientist can view the step's status. This feature will give a summary of steps, connections and connection details from the different levels of refinement of the current step. If the step has several sub-steps and they in turn have sub-steps, the sub-steps are included in the summary.

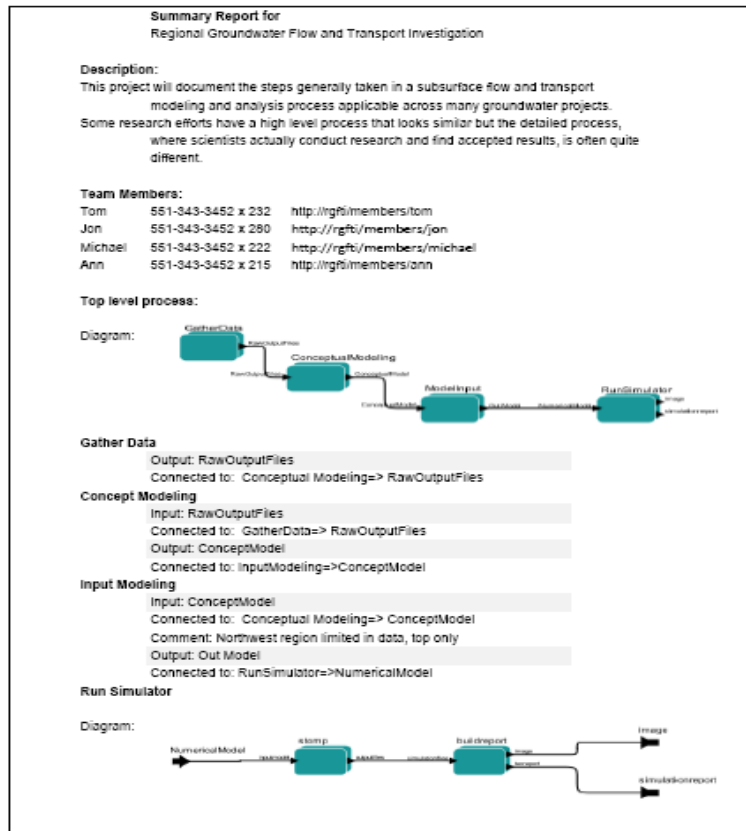


Fig. 8. The first page of a summary report for the RGFTI project. This report gives a summary of the entire project, including project information, e.g., title, team members, process information, steps, and the corresponding knowledge data. This information is provided in a single document.

3.3 The Knowledge-Annotated Data Model

The Kepler source code⁵ and the prototype system are Java implementations. The knowledge-annotated data model for the Kepler prototype is stored in a set of five classes, as shown in Figure 9. The CollaborativeProject class is the focal point for a project: it stores project details, MemberInfo objects, JournalEntry objects and StepInfo objects. There is one StepInfo object for each step defined in a project. The StepInfo class stores JournalEntry objects and PortInfo objects. There is one PortInfo object for each input and each output defined in a step. The PortInfo class stores the annotations (assumptions, constraints, and comments) for an input or output defined in a step. The JournalEntry class stores

⁵ <https://code.kepler-project.org/code/kepler/trunk/modules/build-area>

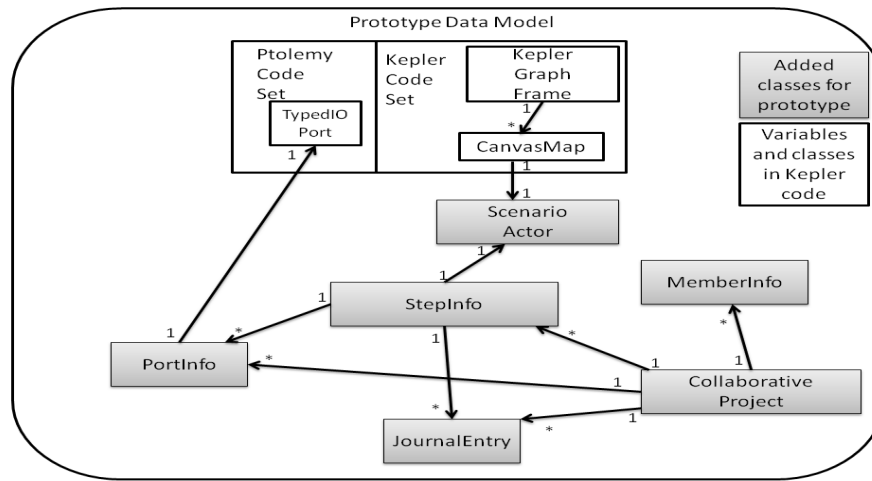


Fig. 9. The knowledge data model for knowledge-annotated scientific workflows in the Kepler prototype. There are five classes used to store all the collected data related to a project.

the annotations entered within the project notes or a step’s research notes. The MemberInfo class stores information about the team members of a project. Figure 9 also describes the canvasMap variable. When Kepler opens up composite actors, their interface and environment are created in a new process space. For the prototype, this caused difficulty in managing the overall annotations within a project. As a result, the canvasMap variable was added to manage the different contexts that would be annotated within a project.

Most workflow tools store their workflows in tool-specific formats, e.g., Kepler stores all workflow information in KAR files. The issue with tool-specific formats is that they limit interoperability and reuse of workflows; e.g., the encapsulated contents of a KAR file would only be readable by KAR-compatible tools. Although we believe it would be unrealistic that all workflow tools conform to the same representational formats, there is no reason why the annotations collected by the prototype can’t be more open to sharing and interoperability. In an effort to enable others to understand what was done and why, the annotations are stored in the Semantically Interlinked Online Communities (SIOC) [5] format. SIOC is an RDF-based OWL model [11]. With this data model, we were able to represent all the annotations collected for a knowledge-annotated scientific workflow project. Our expectation is that this knowledge information, when published on the Web, can be linked with different discussions and comments made by other scientists.

Using the unique identifiers for steps, inputs, and outputs in a Kepler workflow as topic resources within the SIOC model (this is how the SIOC model specifies what the annotation is about) the prototype writes a project’s annotations to

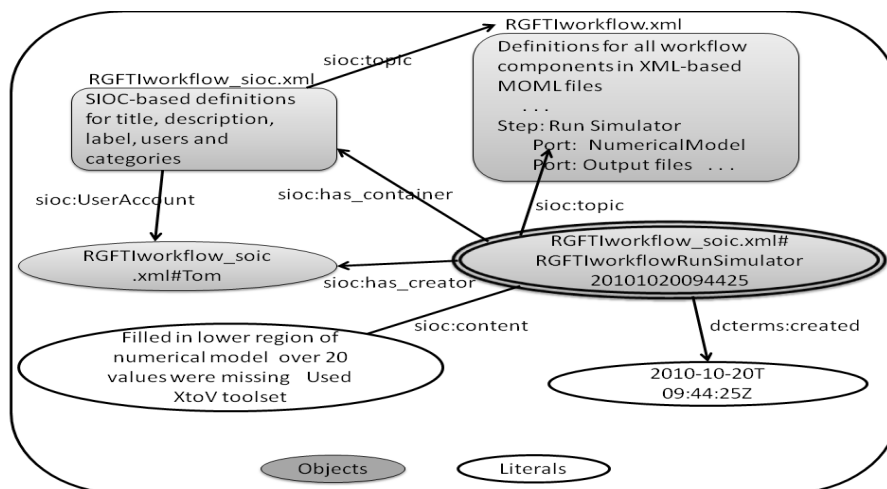


Fig. 10. An RDF graph representation of an SIOC-based description of a comment made in the Run Simulator step. The Run Simulator step is shown in Figure 2

a single SIOC file. Figure 10 shows an RDF graph of the SIOC representation of a comment made about the RunSimulator step’s Numerical Model input. The definition of the workflow’s SIOC model is contained in the RGFTIworkflow_sioc.xml RDF OWL document. There is an entry in this document for each comment, assumption or constraint related to the workflow, a step or a step’s input and output. Following the graph in Figure 10, there is a comment called RGFTIworkflow_sioc.xml#RGFTIworkflowRunSimulator20101020094425 that was created by Tom about the NumericalModel input in the RunSimulator step. This comment is found in the RGFTIworkflow_sioc.xml container. The RGFTIworkflow_sioc.xml container has comments about the workflow in RGFTIworkflow.xml. The SIOC model can be used to describe a variety of details about a comment, including the text in the comment and the creation date.

4 Discussion

Leveraging scientific workflows to document scientific research is enhanced by allowing implicit knowledge annotations to be made during the research process because these annotations reflect why the research was conducted in a specific manner. The prototype presented in this paper has enhanced Kepler by modifying the interface and adding concepts for recording and sharing annotations about the ongoing research. This section discusses issues with the prototype as well as our intentions for future work.

An overriding concern with all information system designs is ensuring that users find the resulting capabilities worth the cost of learning, using, and maintaining information within the system. As with most scientists, the PNNL ground-

water scientists are often extremely busy and thus they are hesitant to use technology if it will slow down their work, make them repeat their work, or impose a rigid process that might limit their work. Our main focus while building this prototype was understanding their process and needs. In fact, an executable workflow is not required to capture process and experimental knowledge for our knowledge-annotation system to be useful. Nevertheless, once a scientist focuses on an executable process and specific data sets, significant complexity cannot be avoided. Because Kepler workflows are executable, the prototype extended the framework to include abstract concepts. Through discussing these extensions with the groundwater scientists we were able to confirm that these features and abstractions align with their research methodology. It is our intention to conduct a more formal evaluation of this tool; taking into account different types of scientific scenarios will help us add characteristics to support a more general scientific audience. This evaluation will also help us understand the details in reporting, collaboration and semantic annotations that will make the collection of this data more useful. For example, currently the knowledge annotations are captured as natural language annotations, yet imposing a controlled vocabulary was considered there was a concern with how much additional work would be required by scientists in conforming to a predefined structure. Understanding the most fitting vocabularies requires further evaluation. Moreover, understanding how to capture knowledge annotations and take advantage of their content without distracting scientists is future work for this research.

Mapping the knowledge annotations to a semantic model, i.e. SIOC, has benefits that we must highlight. For one, the knowledge annotations can be leveraged for reasoning within the existing knowledge of a workflow. Bowers and Ludäscher discuss algorithms for propagating semantic annotations through the knowledge of a workflow in order to infer new annotations [4]. Furthermore, integration of knowledge annotations with the process captured by workflows can be integrated with provenance knowledge that has already been used in semantic based workflow tools [6]. Given the content of the knowledge annotations and the descriptions within the SIOC model, software agents can further evaluate the SIOC content for patterns in terminology or collaborative relationships between scientists. By using semantic structures and publishing them on the Semantic Web, the knowledge annotations have the potential for distribution and integration with the Linked Open Data cloud [3]. As a result, software agents from future research efforts could leverage knowledge annotations from one workflow project as supplemental research data.

5 Conclusions

Understanding the needs of the groundwater scientists involved in the RGFTI project highlighted some key facts for this prototype. First, systems that annotate scientific research should also capture the ongoing implicit knowledge associated with this research. Simply collecting comments and discussions after research is conducted or only capturing executable based knowledge, loses

important information. Second, by leveraging executable scientific workflow systems to add annotations, scientists can embed their notes within an existing framework already built to describe scientific research. However, the workflow environment must be flexible in its ability to collect these annotations. Many scientists must work in an exploratory mode where their process is ad hoc and the tools must support this. Moreover, scientists do not always know the exact steps they will take to conduct their research beforehand and they do not always have the tools they need instrumented as workflow components. Finally, scientists do not always leverage executable components to describe their process: for example, in some cases they construct data models by hand-picking values. What they do use is expertise, and their expertise is important when they are performing research steps and crucial in understanding *how* and *why* research was conducted to produce scientific results.

Executable scientific workflow tools have an advantage when it comes to documenting research; because they are already documenting the execution of scientific analysis, their definition can be leveraged to manage implicit knowledge collected from scientists conducting the research. Unfortunately, current workflow techniques can be confusing and distracting because scientists are forced into a fixed scientific process before they achieve results. This prototype allows for an ad hoc mode of defining scientific process where scientists focus on documenting research through research notes and workflows through abstract concepts such as steps, inputs and outputs. Furthermore, allowing scientists to see comments about data at strategic points within the research process, gives scientists a process-based and team-driven environment for understanding and describing their work.

Collaborations are characterized not just by a sharing of data but also by the entire process and culture by which scientific research is conducted, data is collected, and knowledge is shared, understood, and reused. One barrier to successful, long-term collaborations is the inability to make specific research artifacts and details available. For example, once data has been collected scientists must decide, amongst other details, how to annotate and publish their data so that it can be used by other scientists. Through knowledge-annotated scientific workflows, scientists are documenting the steps they took to perform their research, the correlation between steps, and why the data was created. Publishing this data to a semantic structure means that the structure of the data is well-defined and enabled for reuse. Knowledge annotated scientific workflows simplify the process of annotating scientific workflows with the scientist's notes - knowledge that is necessary for understanding research but not normally collected.

Acknowledgments

This research was partially funded by the DOE SciDAC Scientific DataManagement Center and by the National Science Foundation under CREST Grant No. HRD-0734825. Any opinions, findings, and conclusions or recommendations

expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. EMSL: Capabilities: Subsurface Flow and Transport, 26 January 2011.
2. Kepler KAR: Questions and Answers, 26 January 2011.
3. C Bizer, T Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
4. Shawn Bowers and Bertram Ludäscher. A calculus for propagating semantic annotations through scientific workflow queries. In *In Query Languages and Query Processing (QLQP): 11th Intl. Workshop on Foundations of Models and Languages for Data and Objects, LNCS*, 2006.
5. D. Brickley, S. Stefan, A. Miles, L. Miller, D.O. Caoimh, and C.M. Neville. SIOC Core Ontology Specification. Technical report, 25 March 2003.
6. Paulo Pinheiro Da Silva, Leonardo Salayandia, Nicholas Del Rio, and Ann Q. Gates. On the use of abstract workflows to capture scientific process provenance. In *Proceedings of the 2nd conference on Theory and practice of provenance, TAPP'10*, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.
7. David De Roure, Carole Goble, and Robert Stevens. The design and realisation of the virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561–567, 2009.
8. Deelman E, Gannon D, Shields M, and Taylor I. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 725(5):528–540, 2008.
9. Ora Lassila and Ralph Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 22 February 1999.
10. Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the kepler system: Research articles. *Concurr. Comput. : Pract. Exper.*, 18(10):1039–1065, 2006.
11. Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. Technical report, World Wide Web Consortium (W3C), December 9 2003. Proposed Recommendation.
12. Leonardo Salayandia, Paulo Pinheiro da Silva, Ann Q. Gates, and Flor Salcedo. Workflow-driven ontologies: An earth sciences case study. In *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing*, Amsterdam, Netherlands, December 2006.
13. J. Senker. The contribution of tacit knowledge to innovation. *AI and Society*, 7:208–224, 1993.
14. Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.
15. T. Talbott, M. Peterson, J. Schwidder, and J.D Myers. Adapting the electronic laboratory notebook for the semantic era. In *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems (CTS 2005)*, St. Louis, MO, May 2005.
16. M.D. White and M. Oostrom. STOMP Subsurface Transport Over Multiple Phase: User’s Guide PNNL-15782(UC 2010). Technical report, Pacific Northwest National Laboratory, Richland, WA, 2006.