Noname manuscript No.
(will be inserted by the editor)

CI-Miner: Semantically Enhancing Scientific Processes

Paulo Pinheiro da Silva · Leonardo Salayandía · Aída Gándara · Ann Q. Gates

Received: date / Accepted: date

Abstract The realization of an international cyberinfrastructure of shared resources to overcome time and space limitations is challenging scientists to rethink how to document their processes. Many known scientific process requirements that would normally be considered impossible to implement a few years ago are close to becoming a reality for scientists, such as large scale integration and data reuse, data sharing across distinct scientific domains, comprehensive support for explaining process results, and full search capability for scientific products across domains. This article introduces the CI-Miner approach that can be used to aggregate knowledge about scientific processes and their products through the use of semantic annotations. The article shows how this aggregated knowledge is used to benefit scientists during the development of their research activities. The discussion is grounded on lessons learned through the use of CI-Miner to semantically annotate scientific processes in the areas of geo-sciences, environmental sciences and solar physics: A use case in the field of geo-science illustrates the CI-Miner approach in action.

Keywords cyberinfrastructure · abstract workflow · scientific process · scientific workflow · ontology · distributed provenance

1 Introduction

Cyberinfrastructure (CI) is "the set of organizational practices, technical infrastructure and social norms that collectively provide for the smooth operation of scientific work at a distance" [12]. A goal of CI is to enable novel scientific discoveries through the provision of data with high levels of availability, as well as the complex processing capabilities required for data analysis. As a result of an increasing use of CI in scientific activities, we anticipate that most existing and future scientific systems (i.e., software

P. Pinheiro da Silva, E-mail: paulo@utep.edu
L. Salayandía, E-mail: leonardo@utep.edu
A. Gándara, E-mail: agandara1@miners.utep.edu
A. Q. Gates, E-mail: agates@utep.edu

Department of Computer Science, 500 W. University Ave.
University of Texas at El Paso, El Paso, TX 79968

systems that carry out a process to obtain a result of scientific significance) will need to handle the use of multiple data sources. Different data sources typically will be created through different methods, and they will have different quality assessment practices [20, 21, 41], as well as different format encodings [6, 30]. There have been important advances in CI toward solving the problem of integrating data from multiple sources, as supported by the creation of data centers and virtual observatories such as GEON [1] and EarthScope [11] that serve as data warehouses and that provide standardized tooling and protocols for data retrieval and analysis. As scientific teams continue to adopt CI in their practices, however, alternate ways of collaboration are surfacing that require a decentralized approach to multiple-source data integration [24, 2, 13]. As a result of a new generation of research activities that are more collaborative and multidisciplinary in nature, scientists will need novel ways of collecting data to facilitate sharing, searching, and explaining data and data-derived artifacts.

In general, machines are not always successful at integrating data. For example, it is difficult for a machine to identify that "Benjamin Franklin, Politician" is the same "Benjamin Franklin, Inventor," which may be documented separately in some pages on the Web. Humans may conclude that the different"Franklin" references relate to the same object. A machine, however, without further information, may be unable to determine that relation. Semantic annotations are metadata (i.e., data about data) that allow software applications to identify, for example, that an object in an annotated dataset is the same object that has been semantically annotated in another dataset. In the example above, through the use of semantic annotations, a software application could verify that the two "Franklin" references relate to the same object in the following manner: they are both semantically annotated to be of the type *person*; the "Franklin" part of the term refers to a last name; "Politician" and "Inventor" are position titles; and the "Franklin, politician in Philadelphia" was also the inventor. In general terms, we see that the need for "semantics" is an immediate consequence of the collaborative nature of most scientific processes and the foundation upon which CI-Miner rests.

One challenge that scientists face when working collaboratively is the need to agree on a common and consistent terminology, especially when accessing datasets for scientific analysis. For example, the term "altitude" used in two datasets may be the height in reference to the terrain or in reference to the sea level. Scientists often interpret the meaning through inspection. In other situations, the interpretation of terminology is less obvious. For example, scientists coming from different organizations or fields of expertise may use different data formats and standards to describe similar or related observations. Semantic annotations may be used to describe dataset contents providing a systematic way for machines to verify semantic relationships between dataset attributes, e.g., two attributes are the same, one attribute subsumes another, two attributes are distinct.

Understanding the scientific process used when collaboratively working on scientific analysis is another challenge to consider. To address this challenge, semantic annotations can be used to document the steps taken to perform scientific analysis, as well as to identify the tools and people involved in the creation and execution of the process. These are especially significant in a scientific setting where reliability and traceability of process results is crucial.

This article introduces the CI-Miner methodology for semantically enhancing scientific processes. The methodology, which uses specific notations and is carried out through the use of software tools, is intended for scientists who are not necessarily computer scientists or experts in semantic technologies. Furthermore, scientists who use the CI-Miner methodology recognize that machines and automation should be involved in the process of sharing, searching and explaining scientific artifacts. The methodology considers that scientists have a comprehensive understanding about the processes that they are interested in designing and implementing.

This article describes the methodology by explaining how the semantic annotations added to a scientific process are used to generate explanations on how processes can be executed, how scientific artifacts are derived from experimental data, and how scientists can search, visualize, and ask questions about process results. Section 2 presents background information about how tools, languages, and other approaches are being used to manage and use knowledge about scientific processes. Section 3 introduces a use case to illustrate the challenges of capturing, preserving, and using knowledge about scientific processes. The use case and related challenges are used throughout the article to demonstrate the CI-Miner methodology in action. Section 4 describes the methodology along with supporting notations and tools. Section 5 revisits the challenges presented in Section 3 to highlight the benefits of using CI-Miner. Other case studies are also reported in this section. Conclusions and future developments are presented in Section 6.

2 Background

The CI-Miner semantic enhancements described in this paper are comparable to several ongoing efforts in the research areas of ontologies, workflow specifications, and provenance. This section describes efforts in these areas, including other comprehensive efforts that capture process and provenance knowledge of scientific processes.

2.1 Ontologies to Support Shared Knowledge

Ontologies support a key function of establishing a shared body of knowledge in the form of vocabulary and relationships. A benefit of using ontologies to describe knowledge is that they can be used for a wide range of purposes. For example, the application of ontologies can range from establishing knowledge about general content on the Web, to establishing knowledge about very specialized scientific processes.

Several advances in Semantic Web technology have made using ontologies more feasible. The Ontology Web Language (OWL) [27], a standardized web language for defining ontologies, allows for more interoperability between distinct scientific communities, and hence, ontology development has become a popular activity among scientific communities.

The purpose for developing ontologies, however, varies widely among scientific communities. The TAMBIS ontology [3] and the myGrid ontology [39] are examples of ontologies that are intended to create categorizations of concepts and relationships. For example, TAMBIS categorizes representations of biological structures into “physical” and “abstract.” Additionally, TAMBIS has separate concept divisions for biological processes and biological functions. This notion of distinguishing between the possible representations of a concept helps reinforce the idea that separating concepts into categorizations is beneficial.

The Gene Ontology (GO) [18] is a controlled vocabulary about gene information. It is split up into three main categories, the cellular component ontology, molecular function ontology, and the biological process ontology. In addition to documenting a controlled vocabulary, the purpose of the GO ontology is to document scientific processes.

The Semantic Web for Earth and Environmental Terminology (SWEET) ontologies [37] were developed to capture knowledge about Earth System science. A group of scientists have been capturing several thousand Earth System science terms using the OWL ontology language. There are two main types of ontologies in SWEET: facet and unifier ontologies. Facet ontologies deal with a particular area of Earth System science (earth realm, non-living substances, living substances, physical processes, physical properties, units, time, space, numeric, and data). Unifier ontologies were created to piece together and create relationships that exist among the facet ontologies. Facet ontologies use a hierarchical methodology in which children are specializations of their parent nodes. The SWEET ontologies are currently being used in GEON [15] to capture geologic processes and terms.

Tools that leverage ontologies must facilitate the creation and reuse of ontologies allowing scientists to work together using an agreed upon vocabulary in support of scientific research activities, e.g., creating crustal models of the Earth.

2.2 Workflow Tools to Capture Process Knowledge

Many scientific workflow tools are available and in use for modeling scientific processes. [38] discusses various implementations using such tools within the scientific community, as well as the challenges and benefits of using scientific workflow tools in general. One benefit of workflow tools in general is that they support the capture and preserving of process knowledge by allowing users to build graphical representations of a scientific process. Typically, workflows are built in a systematic way via a user interface and the results are reproducible artifacts that can be reused and modified.

Wings [16] is a workflow tool that allows scientists to specify the steps in a scientific process using semantic annotations in building the workflow. Because the ultimate goal for Wings workflows is to build an executable representation of the scientific process, preliminary work must be done to define the semantic characteristics of the workflow, including dataset and executable components. Having semantic descriptions available during the design phase allows Wings to suggest and verify interoperable components while the workflow is being developed. However, understanding semantic and executable details of components may not be something a group of scientists are prepared to discuss when designing a scientific workflow. In this case, requiring semantically annotated workflow components is an added challenge at an initial stage of workflow design. Allowing workflows to be designed at an abstract level, where scientists are focused on the scientific process without consideration for implementation details until a workflow has been agreed upon can avoid this distraction. Furthermore, it would be helpful to leverage existing knowledge about data and components over the Web as opposed to having to build them locally.

Taverna [42] allows life scientists to build executable workflows over a portal that manages a pre-defined set of data and service components. One restriction of this portal is that, in order to build a workflow, the portal only gives access to components for which it has descriptions. This, similar to Wings, means that scientists need to build a working set of component descriptions before they build a workflow. Given that this portal is focused on life sciences, there is a variety of existing knowledge already built into the portal, but this is not the case for all scientific domains or for scientific teams that choose not to use the portal. Another issue is that the Taverna portal is focused on one scientific domain. There are cases where scientists want access to datasets available from different sources, not just from one scientific domain, i.e., life sciences.

Furthermore, there are additional challenges to using current workflow tools. For example, there are implementations that require modeling of steps that the scientist performs, like instrument calibration or artifact evaluations. Most scientific workflow tools do not support modeling human intervention within scientific processes. Another challenge is that most scientific workflow tools lack an overall methodology for supporting scientists in understanding an abstract scientific process and for refining the information to support sharing and reuse of knowledge and scientific results.

2.3 Tools to Capture Provenance Knowledge

Scientific processes, as defined in this paper, are not necessarily captured by scientific workflows that have executable specifications; however, there is a significant effort of using provenance in scientific workflows that needs to be considered. For example, we observe the use of provenance models in different workflow systems such as REDUX [4], Taverna , Pegasus [23] and Karma [36]. These systems are based on two layers of provenance named *retrospective layer* – information about workflow executions – and *prospective layer* – information about workflow specifications [7]. These retrospective layers tend to have very general concepts for representing process execution traces at the same time that they incorporate domain-specific concepts. Domain-specific concepts can be incorporated in multiple ways; for example, they can be provided by domain-specific ontologies or hard-coded in the systems. In the case of Taverna, which is dedicated to supporting workflows in the Bioinformatics domain, the approach is to include related domain ontologies.

We observe a tendency of provenance systems to develop into comprehensive frameworks for capturing and collecting process and provenance knowledge. The Zoom*UserViews project [8] and the PrIMe methodology to develop provenance-aware applications [29] are examples of such frameworks. Zoom*UserViews is a collaboration among several projects being integrated into a comprehensive solution. Zoom*UserViews focuses on capturing process knowledge in an executable workflow environment. The level of details required for workflow specifications to be executable tends to be distracting for scientists to understand and thus share process knowledge. Many are the benefits of using such framework although captured process knowledge may not be at a level of abstraction that is more convenient for scientists.

The PrIMe methodology uses software engineering practices to elicit "provenance questions" from users, analyze data-generation applications to build data-dependency models that are useful to answer the provenance questions, and instrument the applications with wrappers to capture provenance that can be used to answer the provenance questions. PrIMe centers on the development/instrumentation of software applications from a common understanding to capture provenance. The documentation of scientific processes to promote understanding among a scientific community is not a necessary goal of the methodology.

3 Use Case: Collaboration Challenges for the Creation of Geophysical Studies of Crustal Structure

This section presents a use case in which scientists collaborate to create crustal models of the Earth and share them with others. Crustal models are used to identify the geological structure of the Earth, e.g., the Amazon basin. The identification of such structures is important for several reasons, including to identify mineral reserves and to predict the degree of erosion a region can experience when it loses vegetation coverage, i.e., forest deforestation. Related challenges are described to motivate the need for CI-Miner.

3.1 Use Case Description

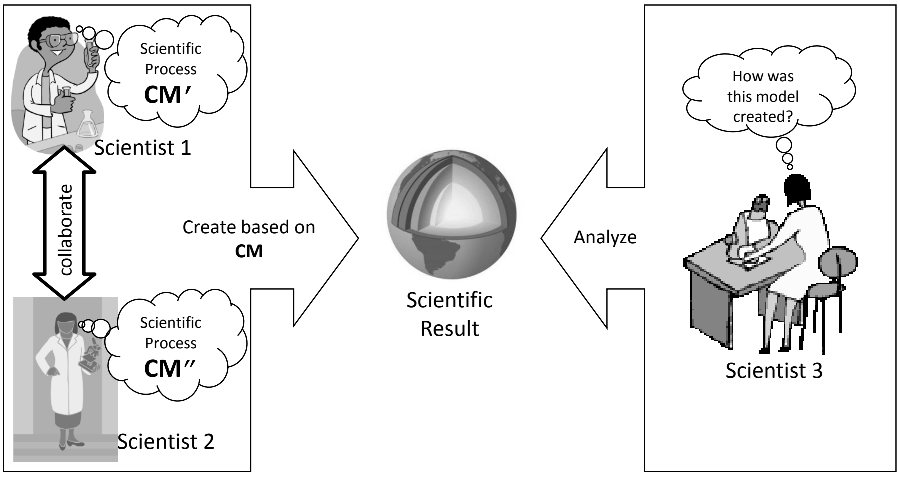


Fig. 1 Creating and reusing a scientific result

Figure 1 depicts a collaboration in which scientists are defining a process to create a crustal model that will be used by another scientist. In this scenario, `Scientist 1` and `Scientist 2` work toward gaining a common understanding about the scientific process `CM` used to create a model of a particular region of Earth. Initially, the understanding of `Scientist 1` about the Crustal Model process ($\mathtt{CM}'$) varies from that of `Scientist 2` ($\mathtt{CM}''$). As the development phase of the process progresses, it is expected that the scientists will eventually reach consensus and document `CM` by describing the main steps of the process along with the flow of information, where the final version of `CM` is based on the original versions $\mathtt{CM}'$ and $\mathtt{CM}''$. The documentation of `CM` is useful to preserve the collaborative effort to reach consensus and to share this knowledge with other scientists. For instance, Figure 1 shows that `Scientist 3` was not involved in the development of process `CM` and the subsequent creation of the crustal model. However, `Scientist 3` wants to understand what is being represented by the crustal model to reuse it in her own work. Depending on the availability of the authoring scientists or supporting documentation about process CM, Scientist 3 may have a difficult time evaluating the scientific result.

The scenario depicted in Figure 1 is a simplification of many scientific collaborations. Complicating factors may include, among many others, scientists collaborating remotely with limited real-time communication, and scientists with different fields of expertise that may use different terminologies to describe scientific processes.

3.2 Use Case Challenges

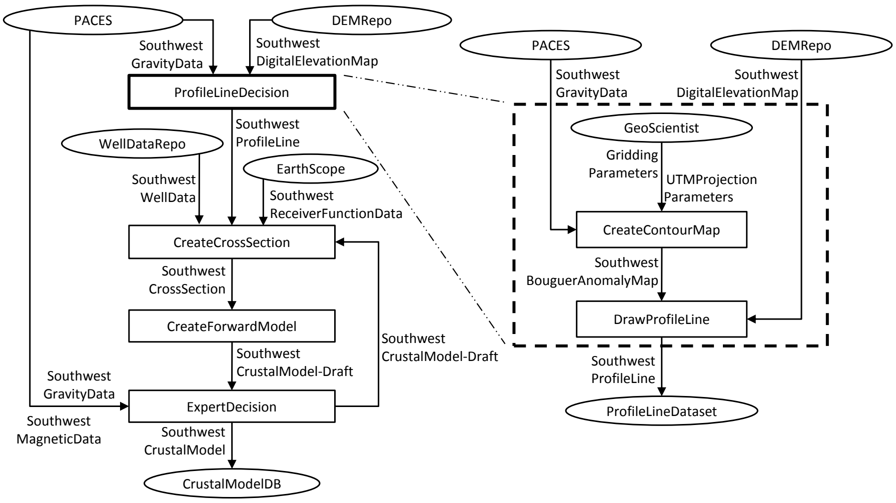


Fig. 2 Process to create a crustal model

Regardless of the complicating factors involved in scientific collaborations, scientists often succeed in developing and using scientific processes to derive products like crustal models. These successes are often hindered, however, when scaling scientific processes for use by a wider community. Lack of mechanisms to capture, preserve, and reuse knowledge about scientific processes are often the cause of the scalability problem. What is more, scientists must share research results and document the processes used to generate research results in a manner that supports their reproduction in order to be successful. The challenges described below are critical because the task of documenting processes requires a significant amount of effort from scientists.

Figure 2 shows the main steps and data flow of the CM process of creating a crustal model, which is used as an example to present the challenges. The process begins with the ProfileLineDecision step at which the scientist determines the profile line of the model. Digital Elevation Maps (DEMs) and gravity data are two cost-effective sources of data that can be used to determine a profile line. The right side of Figure 2 shows additional details about the ProfileLineDecision step of the process. As shown, gravity data has to be treated to create a BouguerAnomalyMap, where input from the geoscientist is required in addition to the gravity data. The step of analyzing DEMs and Bourguer Anomaly Maps to ultimately determine the location of the profile line (`DrawProfileLine`) is a manual step driven by the geoscientist.

The initial steps of the `CM` process describe the selection of specific data sources that scientists decided to include as part of this process. Furthermore, the process describes the need for humans to enter input parameters, as well as the need for humans to analyze data to make decisions.

Challenge 1 – Capturing and preserving process knowledge. *Capturing and preserving human activity in support of scientific processes in a way that offers reliable interpretation by others is difficult. Even for parts of the process that include systematic activities, i.e., machine activity or machine-assisted human activity, differences in scientific terminology across fields of expertise complicates the problem.*

Continuing with the description of the `CM` process, the next step after determining the profile line is to create a cross section about the crustal structure of Earth along the profile line (`CreateCrossSection` in Figure 2). The cross section is comprised of tectonic bodies represented by polygons, where each tectonic body is assigned a density value. In order to construct a credible cross section, the geoscientist needs to research the area of interest to find data that can shed light into the properties of the tectonic bodies. The sources of data used in the process discussed here include: well data, receiver function data, and possibly previously created crustal models that can serve as a basis towards creating a refined model. Because of the diversity of data sources involved, as well as possibly conflicting results obtained from different sources, constructing a cross section requires expert interpretation of the data.

This part of the process involves expert interpretation of various types of data to make decisions about the model to be created. Once a scientific result has been produced, the decisions made at this step may be required to assess its quality. Furthermore, scientists wanting to verify the reproducibility of a scientific result require access to this knowledge.

Challenge 2 – Capturing and preserving provenance knowledge. *The challenge is to capture and preserve provenance knowledge about a scientific result in a way that it can be effectively accessed and used by others.*

The next step in the `CM` process consists of using the cross section created as the basis to construct a forward model of gravity and magnetic data along the profile line. The forward model yields theoretical gravity and magnetic values based on the geometric structures and densities of the tectonic bodies conforming the cross section. The cross section along with the theoretical gravity and magnetic profiles are denominated a `CrustalModel-Draft`.

This part of the process involves the use of a process component that takes some input and systematically constructs a model as output. There may be several systems that can be used to satisfy the requirements of this process component. The scientist may need to make a decision as to which process component is more appropriate.

Challenge 3 – Supporting the integration and interoperation of process components. *The challenge involves capturing and preserving the process-related knowledge required to assess the appropriateness of the system to be used regarding its capability of sharing data with other process components.*

The last step in the process is to compare the resulting crustal model draft against a profile of gravity and magnetic field data observations to determine the fitness level of the theoretical values of the forward model. Based on the criteria of the geoscientist, if the fitness level of the crustal model is good enough, the process ends by having a geoscientist-endorsed crustal model. Otherwise, the process continues by using the crustal model draft as the basis to conduct a refinement iteration.

Challenge 4 – Supporting comprehensive query capabilities for process components and products. *Once a scientific result has been obtained, a challenge is to leverage the scientific process knowledge mentioned in Challenge 1, as well as the provenance knowledge about the result mentioned in Challenge 2 to support advanced query capabilities.*

For example, consider the case where a scientist wants to search for crustal models created specifically with the use of the `PACES` data source illustrated in Figure 2 and where the profile line was determined by `Scientist 1`. Support for comprehensive query capabilitites for process and products could address this problem.

Challenge 5 – Supporting comprehensive visualization capabilities for process components and products. *The challenge is to have access to intermediate results along the execution of the scientific process and be able to visualize them in a way that can shed light into the inner workings of the process.*

Defining scientific processes may involve an iterative process of trial and error, where scientists experiment with different components or steps in the process to determine an optimal choice. Visualization of intermediate results may be a critical capability for scientists to define scientific processes. Similarly, understanding the process inner workings through visualization of intermediate results may also be valuable for other scientists wanting to reuse the scientific process or the products created with it.

4 CI-Miner

CI-Miner offers an approach to help scientists document the knowledge behind their research activities in the form of semantic annotations so that software applications can use this knowledge to better support scientists' research activities. With the use of CI-Miner, scientists can focus on solving scientific problems without worrying about technical nuances of developing and reusing scientific systems. This section presents the methodology behind CI-Miner and the technology used to carry out such methodology.

4.1 Methodology

There are different types of scientific systems that range from legacy to state-of-the-art, well-established to under-development, non-documented to well-documented. In addition, the development of new scientific systems may be required to support novel scientific processes. In order to support scientific processes that leverage these wide ranges of scientific systems, scientists need to understand and be able to communicate the essential functionalities of these systems. In addition, they must be able to communicate the dependencies between these systems and the scientific processes of interest.

The CI-Miner methodology can be applied following an *a posteriori* or an *a priori* approach to documentation. In an *a posteriori* approach (i.e., from system to documentation) the intention is to analyze the inner workings of an existing scientific system in order to create systematic documentation that can be used to understand its appropriateness to a particular scientific process, and that can be used to enhance the existing scientific system with features that show (from the perspective of the scientist) the steps that are carried out by the scientific system to produce a scientific result.

In an *a priori* approach (i.e., from documentation to system) the intention is to systematically document the scientific process from the perspective of the scientist in order to support the development of a corresponding scientific system, or to identify existing systems that can be reused. As in the *a posteriori*, the systematic documentation produced can be used to enhance the system to be developed or reused with features that show (from the perspective of the scientist) how scientific results are produced using the system.

It is assumed that scientists adopting CI-Miner are capable of capturing common knowledge about the scientific process at hand. It is also assumed that once scientists reach a common understanding about the process, that they can establish the fundamental concerns of carrying out the scientific process, and that they can evaluate whether a scientific system (or parts of it) appropriately support the concerns. For example, a geoscientist that is adopting CI-Miner and that is tasked with the scientific endeavor of creating a crustal model (i.e., the case study of Section 3) should be able to establish that creating a forward model of theoretical gravity values from a cross section is a necessary step. Furthermore, the geoscientist should be able to evaluate whether a given forward modeling software provides adequate functionality for that step.

The methodology is presented as a series of steps that a scientist needs to accomplish in order to systematically document a scientific process, as well as to enhance scientific systems that are used to support the scientific process. The order of the steps may vary, for example, depending on whether an *a posteriori* or an *a priori* approach is used.

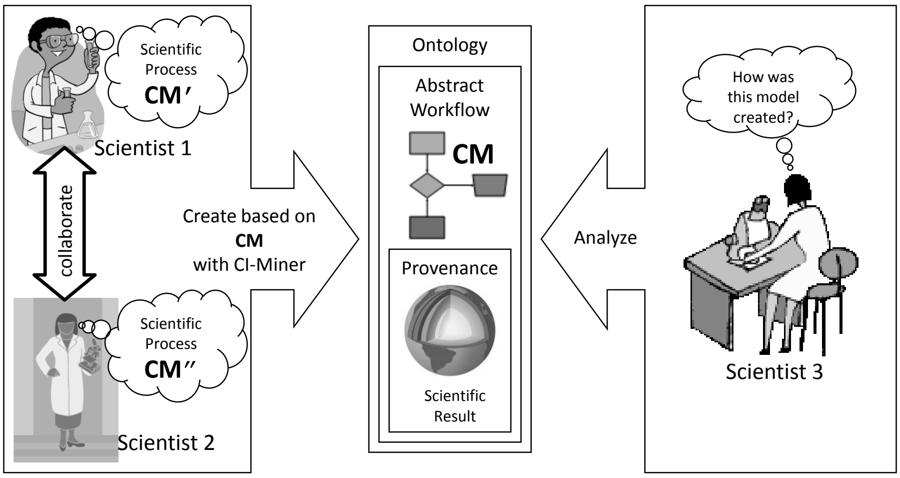


Fig. 3 Scenario about creating and reusing scientific results with CI-Miner

Figure 3 shows the scenario presented in Section 3, along with the semantic annotation resulting from using the CI-Miner methodology (the annotations are represented by the boxes around the scientific product in the center of the figure). The scientific result is presented along with provenance semantic annotation (i.e., the `Provenance` box around the scientific product) that encodes the knowledge about how that result was created, which data sources where used in its creation, and what human decisions where made towards creating it. Furthermore, the provenance about the scientific product is grounded upon a documented process `CM` that is encoded in the form of an abstract workflow. In CI-Miner, a scientific product of interest is called the **Subject of Discourse (SOD)** of the methodology. Further, the scientific process responsible for the derivation of a SOD is the **Process of Discourse (POD)** of the methodology. Scientists may use the methodology to documents multiple SODs and corresponding PODs including PODs that can derive multiple SODs each. An assumption, however, is that a scientist may only focus on one SOD and one POD at a time while using the CI-Miner methodology.

In Figure 3, the fact that the `Abstract Workflow` box is around the `Provenance` box means that there is a mapping between the provenance knowledge concepts and the abstract workflow concepts, and that this mapping is encoded in the provenance allowing one to reach the corresponding abstract workflow by inspecting the provenance. Lastly, both the abstract workflow of `CM` and the provenance of the scientific result are grounded on common terminology that is described in the form of an `Ontology`. The ontology and abstract workflows are useful in the collaborative phases between `Scientist 1` and `Scientist 2` to defining process P to create the scientific result. The provenance about the scientific result is useful to `Scientist 3` because the scientific result is accompanied with additional information that describes how that scientific result was created.

Step A: Establish a vocabulary of terms about the process of discourse. The vocabulary defined in this step is the knowledge represented by the `Ontology` box in Figure 3[1].

1. Identify and name the kinds of data that are used in the POD. These include things such as datasets, input parameters, field observations, and data logs. It is important to emphasize that the goal of this step is to identify *kinds of data* rather than proper data. For example, a scientific process about the creation of a gravity contour map may use multiple gravity datasets, where each dataset contains gravity readings about a distinct region. In the case of these datasets, only one kind of data needs to be identified and named – "gravity data." The kinds of data identified in this step are the data concepts of the POD;
2. Identify and name the SOD that is the main outcome of the POD. The SOD is considered also a data concept of the POD. A POD may derive other products in addition to the SOD. The scientist may choose to include or disregard these additional data concepts;
3. Identify and name the kinds of methods that are used in the process. These methods are process components such as software tools and human activities that take some data as input and transform it. Similarly as before, the goal of this step is to identify *kinds of methods*. For example, there may be several tools capable of computing

[1] The vocabulary defined in this step may build upon other existing vocabularies documented as ontologies. This is referred to as *ontology harvesting*.

the standard deviation of a dataset attribute. In this case, only a generic method needs to be identified and named – "standard deviation." The kinds of methods identified in this step are the method concepts of the scientific process;

4. When appropriate, compare the data and method concepts identified thus far to other established vocabularies in the field of study to refine the process vocabulary, i.e., change an identified concept name for a concept name that is more established in the scientific community, or identify synonyms. For the case of data concepts, harvesting concepts from other established vocabularies is useful for purposes of data integration. For example, a data concept initially identified by a scientist as *Corrected Gravity Data* may be compatible to the definition of *Processed Gravity Data* used in the vocabulary endorsed by an established organization that maintains a data repository of gravity datasets; hence, those datasets could be used in the execution of the scientific process.
5. Identify the input and output relations between the data and method concepts of the POD. In other words, for each method concept identified in A.3, identify the data concepts that the method concept consumes, i.e., input data, and each data concept that the method concept produces, i.e., output data.

Step B: Specify an abstract workflow specification for the process of discourse. The specification defined in this step is represented by the `Abstract Workflow` box in Figure 3.

1. Model the POD as an abstract workflow in terms of the vocabulary established in Step A. By using the data and method concepts previously established, the scientist models his or her understanding of the scientific process as a set of steps, each of which is represented by a method, and each of which requires a set of input data and results in a set of output data. The set of steps is interconnected through data interdependencies. For example, Figure 4 shows an abstract workflow specification where data is represented by directed edges, and methods are represented by rectangles. The `ProfileLineDecision` method has the data `SouthwestGravityData` and `SouthwestDigitalElevationMap` as input, and the data `SouthwestProfileLine` as output. Furthermore, the method `CreateCrossSection` has among its input data concepts the concept `SouthwestProfileLine` that is the output of `ProfileLineDecision`, hence effectively specifying a data dependency between these two steps in the workflow.
2. Revisit Step A to identify additional terms that may not have been identified during the initial analysis of the POD, but that the scientist may have identified as he (she) started to model his (her) understanding of the scientific process as an abstract workflow.
3. Verify that the SOD identified in A.2 is encoded as the final outcomes of the abstract workflow. Scientific products other than the SOD can be added to abstract workflows.

It should be noted that in the Step B of the methodology, the data and method artifacts used to build the workflow correspond to instances of the data and method concepts defined in Step A. This means that one or more instances of a concept can be employed in the abstract workflow. For example, Figure 4 shows two instances of the `GravityData` concept, one is used as an input to `ProfileLineDecision`, while the other is used as an input to `ExpertDecision`.

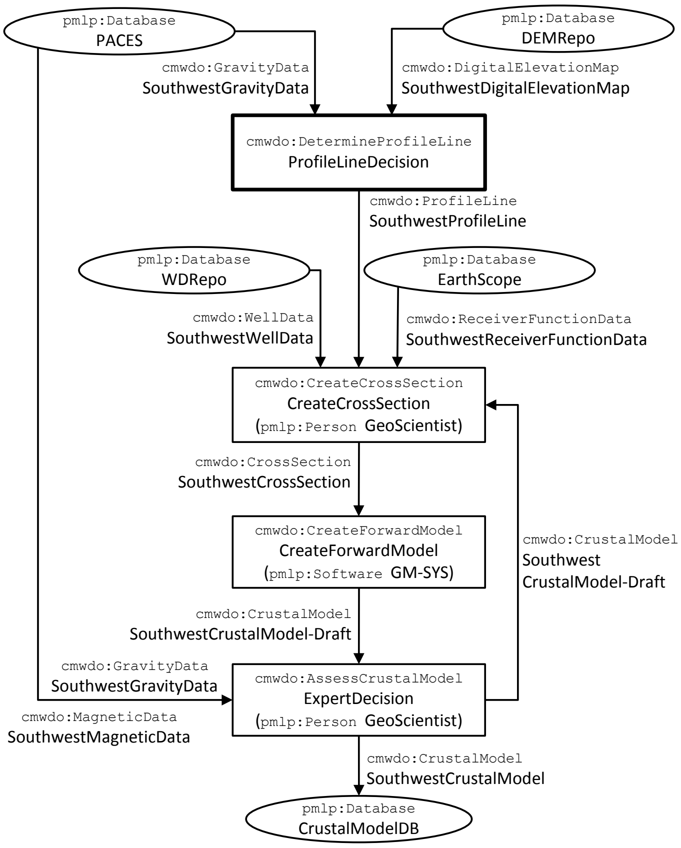


Fig. 4 Semantic Abstract Workflow about the process of creating a crustal model. Rectangles represent the steps of the process, directed edges represent data, and ovals represent data sources and sinks. The labels are of the form xxx:yyy zzz, where xxx is the ontology that defines the concept, yyy is the name of the concept, and zzz is the name of the of the particular instance. Optionally, the labels of some steps include another instance name between parenthesis that represents the agent responsible to carry out that step.

The design of abstract workflow specifications can require more elaborated abstraction mechanisms. For example, the workflow diagram illustrated in Figure 2 shows an expansion of the `ProfileLineDecision` rectangle, which provides additional details about that step. The supported mechanisms to model workflows at multiple levels of abstraction are described in [14].

Step C: Instrument scientific system to capture provenance about the subject of discourse. The knowledge about process executions captured in this step are represented by the `Provenance` box in Figure 3.

1. Map the methods (or steps) identified in the abstract workflow to scientific systems or parts of scientific systems that correspond to such steps. In the case of using the methodology with an *a posteriori* approach, this mapping should be fairly straight forward, since the terminology and abstract workflows defined in the previous steps

are modeled based on the analysis of the scientific system. In an *a priori*, however, the mapping may be more challenging since it may involve cases where a scientific system does not exist for a corresponding step. For example, these steps may be due to *human interventions* to the process that are important to be recognized as steps in scientific processes;

2. Identify metadata to describe the processing of each step of the abstract workflow, as well as the metadata to describe the data consumed and the data produced by each step. For example, the `CreateForwardModel` step presented in Figure 4 could include metadata such as the name of the scientific system used to perform the step, e.g., GM-SYS. The `SouthwestCrossSection` data consumed by that step could include metadata such as the name of the region, e.g., Southwest;
3. For steps that are not classified as human interventions, use *data annotators* to capture provenance. Data annotators use the metadata identified in Step C.2 to annotate the provenance for the step execution. With the use of the input/output data contained in the abstract workflow for each step of the POD, data annotator modules are automatically generated for each step in the form of template code that captures the inputs used when a step is initiated, as well as the outputs resulting when a step is finished;
4. *[Optional]* For steps that are human interventions, consider the creation of tools that would enable scientists to document their interventions;
5. Inspect the abstract workflow specification for properties that dictate when provenance should be logged. For example, when intermediate artifacts do not persist during execution of the workflow, an in-processing approach must be used to capture the provenance for intermediate artifacts and, if needed, to capture the artifacts themselves before they are expunged from the process;
6. Modify the system coordinating agent to invoke data annotators. This implies that the invocation of data annotators need to occur at precise moments in execution, thus the coordinating agent of the data annotators is also the coordinating agent of the concrete workflow. Deciding where to add these calls to a workflow requires that a user understands specifics of a concrete workflow, such as which parts correspond to the coordination of process (i.e., control flow) and which parts correspond to the execution of workflow activities; for it is this knowledge that is needed to instrument the workflow;
7. Execute the scientific process;
8. If a workflow does not delete intermediate results, or if users are unable to modify a workflow, then a non-invasive post-processing annotation can be used. In this case, knowing about workflow how/when/where workflow activities are invoked is less important than knowing specific properties of data output from the activities. This is because post-processing annotators search for the existence of certain types and properties of data to clue in that a particular workflow activity was executed. For example, if an annotator was configured to capture provenance associated with the crustal model activity `CreateForwardModel` it would search the file system for the existence of a `SouthwestCrossSection`, which would provide evidence that the `CreateCrossSection` was executed;
9. *[Optional]* In the absence of a comprehensive search capability for semantic annotations published on the Web, i.e., a search engine that can locate and index semantic annotations, identify the location where provenance documents are initially stored. This step may be also required if standard search capabilities are insufficient for selecting provenance-related search criteria;

10. *[Optional]* In the absence of a comprehensive search capability for semantic annotations published on the Web, create a routine for crawling the provenance documents and for storing them in a triple-store database.

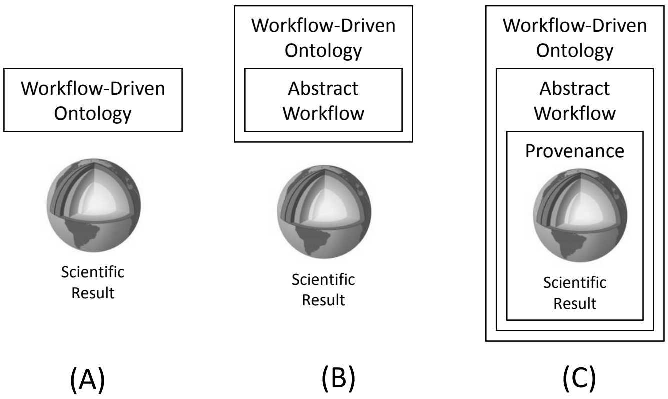


Fig. 5 Outcomes after each step of the methodology. (A) After Step A the outcome is a workflow-driven ontology that captures concepts related to the process of discourse (POD). (B) After Step B the outcome is an abstract workflow of the POD to create the subject of discourse (SOD), where the abstract workflow is grounded on the concepts defined in the ontology of Step A. (C) After Step C the outcome is a searchable provenance artifact that is linked to the SOD.

Figure 5 shows the outcomes that a scientist obtains after each of the phases of the methodology. After Step A the scientist has established a basic vocabulary to talk about the POD. This vocabulary is documented in the form of an ontology, i.e., a workflow-driven ontology. After Step B the scientist has created an abstract description of the POD in terms of the vocabulary defined in the previous step. After Step C the scientist has a collection of provenance documents for each execution of the POD, where each of these provenance documents is linked to its corresponding SOD.

The next section describes how these steps are accomplished with the use of ontology encoding, abstract workflow specification, and provenance, as well as with the use of tools for managing these encodings.

4.2 Implementation – Supporting tools and notations

This section explains how CI-Miner is implemented through the use of a collection of tools and notations.

4.2.1 Ontology Support

In computer science, ontologies have gained a lot of attention since the Semantic Web initiative was revealed in 2001 [5]. Ontologies are artifacts used for capturing knowledge about a given domain in terms of concepts and relationships among concepts. The way ontologies are created and used is driven by their purpose of use. In particular, Guarino

[19] suggested the classification of ontologies according to their level of dependence to a particular task or point of view.

In the CI-Miner case, the focus is on the use of *task ontologies* for capturing knowledge about a scientific discipline through the use of process-related concepts. This approach is called workflow-driven ontologies (WDOs) [35]. The two main classes of WDO upper-level ontology[2] are *Data* and *Method*. The *Data* class is representative of the data components of a scientific process. These can be things such datasets, documents, instrument readings, input parameters, maps, and graphs. The *Method* class is representative of discrete activities involved in the scientific process that transform the data components. The intention of WDOs is to allow scientists to capture process-related concepts by extending the hierarchies of *Data* and *Method* (Steps A.1 and A.3 of the methodology). Furthermore, *Data* and *Method* classes can be related through *isInputTo* and *isOutputOf* relations to capture their data-flow interdependencies with respect to a scientific process (Step 5 of the methodology).

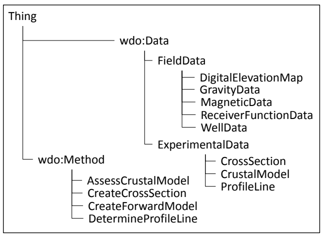


Fig. 6 Crustal Modeling Workflow-Driven Ontology

Figure 6 shows a taxonomic representation fragment of the Crustal Modeling Workflow-Driven Ontology (`cmwdo`). As shown, `cmwdo` is a WDO because the taxonomy is grounded on the `wdo:Data` and `wdo:Method` classes. According to `cmwdo`, both `DigitalElevationMap` and `GravityData` are `FieldData` that may be used in a scientific process (because `FieldData` is of type `wdo:Data`). Moreover, in terms of process functionalities, `cmwdo` shows that a scientific process used to develop crustal models, i.e., `CM`, may have a step called `CreateForwardModel`, since it is a subclass of `wdo:Method`. As `cmwdo` shows us, these are the terms that geoscientists may use to describe the process of building crustal models. It is important to note that the description of a crustal modeling scientific process is not in `cmwdo`: the ontology does not know how many data instances and method instances are needed to implement the process and how these data instances and method instances are connected to produce crustal models. WDO-It![3] [32] is a tool that enables scientists to create workflow-driven ontologies for their area of interest.

4.2.2 Workflow Support

From a scientist's perspective, processes can be generalized as graphical structures containing the following: nodes representing discrete activities, and directed edges representing data flow between those activities. Activities connected through edges effec-

[2] http://trust.utep.edu/1.0/wdo.owl

[3] WDO-It! tool available at http://trust.utep.edu/wdo/downloads/

tively determine data dependencies between the activities. Information fed into the process is provided by data sources, and information generated by the process may be stored in a data sink. Traversing the graph from its initial data sources to its final data sinks simulates the action of carrying out a complex process conformed of simpler activities. To deploy such a representation of a process as an automated or semi-automated system, additional control flow information is necessary to determine the rules that guide the graph traversal.

According to the CI-Miner approach, scientific process specifications can be captured by semantic abstract workflows (SAWs). SAWs are artifacts capable of describing the process at a level of detail that is adequate for scientists. The term "semantic" refers to the fact that nodes correspond to instances of `wdo:Method` and edges correspond to instances of `wdo:Data`, as captured in the WDO used to create a SAW. "Abstract" refers to the fact that the captured process lacks additional constructs necessary to produce automated systems that, for example, would implement the modeled process as a scientific workflow. In this sense, SAWs are not committed to be executable workflow specifications.

Figure 4 shows an example of the graphical notation of SAWs for our use case. Instances of `wdo:Data` are represented by directed edges and instances of `wdo:Method` are represented by rectangles. Data and methods instances are labeled with a name given by the scientist and prefixed with the name of their corresponding user-defined WDO class. SAW's *Sources* and *Sinks* are introduced in the graphical notation of SAWs as a bootstrapping mechanism to indicate the starting and ending points of a process, and these are represented by ovals. Sources and Sinks are also labeled with the name of their corresponding class defined in the provenance component of the Proof Markup Language (PML-P) ontology discussed below.

WDO-It! can be used to build SAWs from WDOs (Step B.1 of the methodology). By dragging and dropping WDO data and methods inside a graphical workspace, scientists can instantiate methods and use data to connect methods. SAWs do not have the capability to model control flow. This may be beneficial in that it removes a layer of complexity for the scientist. For scientists who are more engaged in the design process, however, this restriction may introduce a level of frustration, for instance, to include information such as the number of times a method iterates or the conditions for the execution of methods. Despite these limitations, the benefits of SAWs lay in their simplicity to describe scientific processes, as well as to include additional information related to provenance as described in the following section.

4.2.3 Provenance Support

Once a SAW has been authored, e.g., using the WDO-It! tool, it can be used to drive the generation of "data annotators" that are modules designed to capture provenance associated with workflow activities (Step C.3 of the methodology). Executing a set of data annotators corresponding to a single SAW is similar to executing a workflow in the sense that some coordinating agent is needed for both the synchronized invocation of each data annotator and for the message passing facilities needed for communicating between them.

Data annotators are built for the main purpose of logging provenance; they are not allowed to transform data belonging to the scientific process of interest. Therefore, data annotators use provenance as the exclusive language for communication (i.e., the inputs and outputs of data annotators are provenance elements). When using data annotators, provenance is transformed by each annotator by always enhancing the input provenance trace with more information.

The provenance captured in CI-Miner is encoded in the Proof Markup Language (PML) [28,31], which is designed to support distributed provenance; thus, data annotators can too be distributed along with any remote services that are invoked by a workflow. This is possible because the inputs to data annotators, which are PML node sets associated with executions of the dependent workflow activities, are referenced by URIs. This is convenient because often times complex scientific processes are modularized and controlled by a master script that in turn makes calls to services which may or may not be located remotely. In these cases, the agent coordinating the data annotators does not need to know about provenance as a whole, but only encounters the URIs of intermediate provenance elements.

The goal of capturing provenance about data is to support the explanation of how data is created or derived, e.g., which sources were used, who encoded the data, and more. As shown in Figure 7, the PML ontology defines primitive concepts and relations for representing provenance about data. PML is divided into two modules [25][4]:

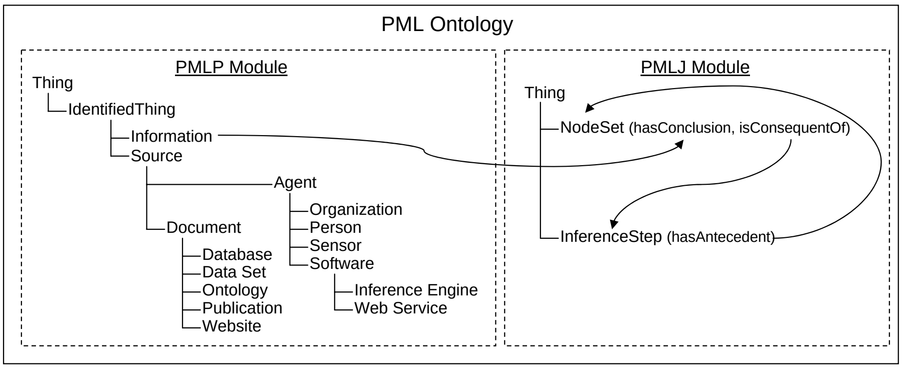


Fig. 7 A simplified view of the PML Ontology

- The *justification module*[5] (PML-J) defines concepts and relations to represent dependencies between identifiable things;
- The *provenance module*[6] (PML-P) defines concepts to represent identifiable things from the real world that are useful to determine data lineage. For example, sources such as organization, person, agent, service, and others are included in PML-P.

The goal of the justification ontology is to provide the concepts and relations used to encode the information manipulation steps used to derive a conclusion that in a scientific process is often a dataset. A justification requires concepts for representing conclusions, conclusion antecedents, and the information manipulation steps used to transform/derive conclusions from antecedents. Although these terms stem from the theorem proving community, they can be mapped to more familiar workflow terms;

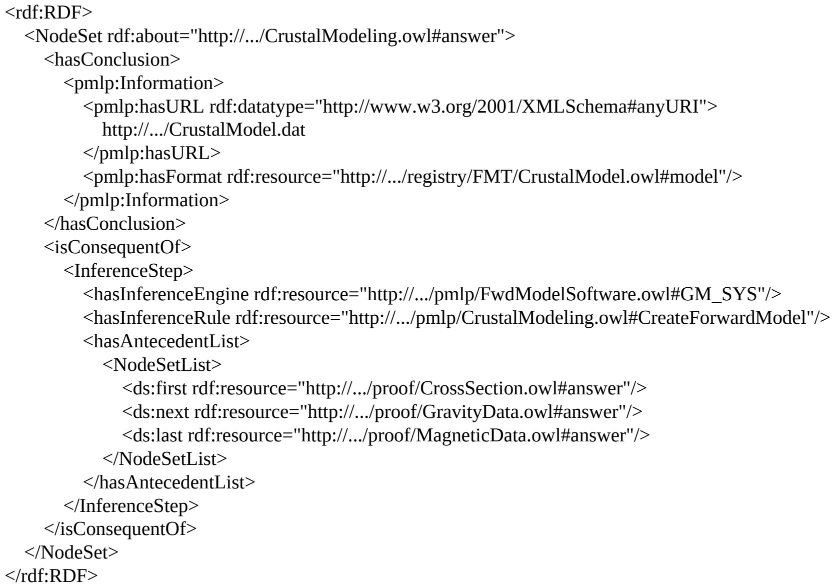
[4] The ontology includes a *trust relation module* that is not used by CI-Miner.

[5] http://inference-web.org/2.0/pml-justification.owl

[6] http://inference-web.org/2.0/pml-provenance.owl

for example, conclusions refer to intermediate data and antecedents refer to the inputs of some processing step. The justification vocabulary has two main concepts: `pmlj:NodeSet` and `pmlj:InferenceStep`. A `pmlj:NodeSet` includes structure for representing a conclusion and a set of alternative `pmlj:InferenceSteps` each of which provides a distinct justification for the conclusion. The term `pmlj:NodeSet` is chosen because it captures the notion of a set of nodes (with inference steps) from one or many proof trees deriving the same conclusion. Every `pmlj:NodeSet` has exactly one unique identifier that is web-addressable, i.e., a URI.

Figure 8 outlines a PML node set capturing the processing step implemented by an instance of `cmwdo:CreateForwardModel` in Figure 4. The output of `cmwdo:CreateForwardModel` is an instance of `cmwdo:CrustalModel` called `SouthwestCrustalModel--Draft`, and this data is captured in the *Conclusion* element as an instance of `pmlp:-Information`, as described below. Additionally, the inputs consumed by `cmwdo:CreateForwardModel` are captured as *Antecedents* of the node set's inference step.



```
<rdf:RDF>
  <NodeSet rdf:about="http://.../CrustalModeling.owl#answer">
    <hasConclusion>
      <pmlp:Information>
        <pmlp:hasURL rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
          http://.../CrustalModel.dat
        </pmlp:hasURL>
        <pmlp:hasFormat rdf:resource="http://.../registry/FMT/CrustalModel.owl#model"/>
      </pmlp:Information>
    </hasConclusion>
    <isConsequentOf>
      <InferenceStep>
        <hasInferenceEngine rdf:resource="http://.../pmlp/FwdModelSoftware.owl#GM_SYS"/>
        <hasInferenceRule rdf:resource="http://.../pmlp/CrustalModeling.owl#CreateForwardModel"/>
        <hasAntecedentList>
          <NodeSetList>
            <ds:first rdf:resource="http://.../proof/CrossSection.owl#answer"/>
            <ds:next rdf:resource="http://.../proof/GravityData.owl#answer"/>
            <ds:last rdf:resource="http://.../proof/MagneticData.owl#answer"/>
          </NodeSetList>
        </hasAntecedentList>
      </InferenceStep>
    </isConsequentOf>
  </NodeSet>
</rdf:RDF>
```

Fig. 8 Example of a provenance encoding in PML

Figure 7 shows that the foundational concept in PML-P is `pmlp:IdentifiedThing`, which refers to an entity in the real world. These entities have attributes that are useful for provenance such as name, description, create date-time, authors, and owner. For example, in Figure 8 the node set is adorned with PML-P instances that effectively convey that this node set corresponds to an execution of `cmwdo:CreateForwardModel`. The PML-P *inference engine* instance is named `GM-SYS` to indicate that this captured step is in fact an execution of a software system called “GM-SYS”. Furthermore, the PML-P *inference rule* instance describes the specific step, (e.g., `CreateForwardModel`) in terms of what the step does and what organization is responsible for this particular implementation of the algorithm. PML includes two key subclasses of `pmlp:IdentifiedThing` motivated by provenance representational concerns: `pmlp:Information` and `pmlp:Source`. The concept `pmlp:Information` supports references to information at various levels of granularity and structure. The concept `pmlp:Source` refers to an information container, and it is often used to refer to all the information from the container. A `pmlp:Source` is further specialized into a `pmlp:Agent` or a `pmlp:Document`, and a document can be a database, a dataset, and a publication among others. PML-P provides a simple, but extensible taxonomy of sources.

5 CI-Miner Benefits and Discussion

To demonstrate some of the benefits of our methodology, we discuss the use of CI-Miner in a collection of case studies. To facilitate this discussion, we use the five challenges identified in our use case.

5.1 Process Knowledge Preservation

The main goal for steps A and B of the CI-Miner process is to generate documentation of a scientific process in a form that can be understood by a diverse group of scientists. We have implemented the CI-Miner process in various projects and, as a result, have created accompanying ontologies and abstract workflows representing process knowledge. Many of these implementations involve legacy systems, software systems already implemented to perform the automated steps of scientific research activities. Initially, we found that many of the discussions of the processes behind legacy-based systems focused on source code. These discussions occurred between a scientific team of non-programmers and a few programmers and the discussions would often break down. At times, there was little understanding as to what the code was doing, regardless of a scientists exposure to the overall process. Discussing abstract workflows built from common terminology avoided the distractions resulting from discussions of variables, source code and program syntax. One interesting observation was when two scientists, who had been working together for years, had fundamental disagreements as to how a process worked. By creating reproducible representations of scientific activities, we were able to facilitate agreement in process, addressing **Challenge 1** in the use case. In this case, the benefit of creating abstract workflows was a consistent understanding of a single process understood in two different ways. In many scientific processes, it was necessary to document both the steps performed by software and those performed by a scientist. Some workflow tools will not allow for such differentiation, e.g., all steps must be machine executable. Working at an abstract level within a workflow allowed the capture of scientific process knowledge, regardless of implementation details. One final benefit of the CI-Miner step to capture process knowledge comes from using semantically annotated technology. Many projects, with which we worked, needed to integrate datasets available over the Web. For example, the Virtual Solar-Terrestrial Observatory (VSTO) ontology [7] is being used by one project to describe a process for capturing images of the sun. By using ontologies to describe inputs, we were able to harvest the terminology within the VSTO ontology and use it in the project's workflows. In this way, the workflows reuse terminology from a trusted and accepted source.

[7] http://dataportal.ucar.edu/schemas/vsto.owl

5.2 Provenance Knowledge Preservation

An important benefit to building abstract workflows is the direction workflows give to provenance preservation. Using the technologies available to CI-Miner, in particular PML, we were able to build deposits of provenance information at data sources. For example, PML-P nodes were made to annotate the characteristics of data reading instruments and the nodes were published for access over the Web. Whenever data is accessed from these instruments, the related PML-P node provides an explanation. Within the semantically annotated abstract workflows of CI-Miner, source information can be shown as inputs and outputs of components. With specific characteristics about the source, we are able to understand workflow components within the context of the workflow, not just at the distributed web location. Given that the workflow captures the steps within the process and has access to knowledge about that the process, it can automatically generate a script that would direct the provenance capture from inputs and outputs and to build data annotators. The annotators facilitate the collection of source information when the scientific process is actually executed. By following the CI-Miner methodology and using the combination of tools available, e.g., abstract workflows, source information and data annotators, a significant amount of provenance has been built into the real-time collection of data for a scientific process involving solar physics. In an environmental study, we needed to annotate data that was produced days, weeks, and even years ago. As a result, there were thousands of data files so a manual approach to annotating them was unrealistic. To help with this challenge, the workflow was used to understand the overall process of capturing the data, and post-processing annotators were built to annotate the data. The overall result, whether using data annotators or post-processing annotators, is the aggregation of provenance to the scientific artifacts. Moreover, the preserved provenance is in a structured format, machine readable and available for access over the Web, resulting in searchable descriptions of data that can be used by other scientists to understand the results. The accomplishments described above address **Challenge 2** in the use case.

5.3 Data Integration and Interoperability Capabilities

Data integration is facilitated by WDOs, SAWs, and ontologies harvested by WDOs. The data hierarchy in a WDO provides an explicit way of annotating whether the content of two data sets have the same kind of measurements, and SAWs identify where these datasets are used in the process. For example, Figure 4 shows the CM use of a dataset called `SouthwestGravityData`, which is of type `cmwdo:GravityData`. This means that the CM process can be repeated for other regions of the planet as long as the dataset used in this step of the process is about gravity measurements in the new region of interest and the dataset is of type `cmwdo:GravityData`. Furthermore, let say that a scientist decides to produce crustal models for an extensive area of the U.S., e.g., the western part of the U.S. In this case, the `ProfileLineDecision` step in the CM process could be based on a new dataset of type `cmwdo:GravityData` derived from the merging of the content of `SouthwestGravityData` and `NorthwestGravityData` datasets.

A more complex data integration scenario is when datasets are not of the same type, but still need to be integrated. For instance, according to Figure 6, `cmwdo:GravityData` is a specialization of `cmwdo:FieldData`. To execute the `ProfileLine-`

`Decision` step for the western part of the U.S., let's assume that the `NorthwestGravityData` is of type `cmwdo:FieldData`, but not of type `cmwdo:GravityData`. This means that some of the attributes in the two datasets are the same, which is why both datasets are of type `cmwdo:FieldData`), and others are different, which is why only `SouthwestGravityData` is of type In this case, data integration may be accomplished by inspecting the specifications of the fields of these datasets. These specifications may be available in the ontologies describing these datasets and the ontologies harvested by WDO. Thus, through these inspections, tools can perform semantic matching [17] to verify how terms are pair-wise related and if the datasets can be combined. If such a description is unavailable, scientists would need to analyze the datasets manually and determine how their concepts match semantically. Even in this situation, there is still benefit in using WDOs and SAWs to document scientists' findings that would enable future data integration efforts. The semantic enhancements described above are essential steps towards a systematic data integration approach based on CI-Miner. These enhancements address **Challenge 3** in the use case.

In terms of interoperability, OWL is the language on top of which WDOs, SAWs and PML documents are built. Using OWL terminology, WDOs are OWL documents that capture ontological classes and relations, while SAWs are OWL documents that capture knowledge based on the knowledge captured in the WDOs. Hence SAWs do not contain class or property definitions, but instead, include only instances of the classes and properties defined in WDOs. With this knowledge about OWL, we can return to the use case.

5.4 Search and Query Capabilities

Many computationally expensive processes are often repeated multiple times because of the difficulty scientists may have to search for process results whether they are published on the Web or stored in a local file-system. For instance, the scientist using the crustal modeling SAW needs to decide a profile line for the southwest part of the United States. To make this decision, according to Figure 2, the scientist may need to use a `BouguerAnomalyMap` about the southwest region of the U.S. However, if Bouguer Anomaly maps are published on the Web, chances are that no search engines can locate them just with the use of keywords.

Without the use of semantic annotation, search engines are limited on how much knowledge they are capable of extracting out of ordinary Web content [26], i.e., non-annotated web content. If Bouguer Anomaly maps are semantically annotated, for example, with the use of CI-Miner, and published in a place that can be indexed by a semantic-aware search engine such as IWSearch [33] and Swoogle [10], then the scientist may be able to locate the appropriate map.

The characteristics of the request in the above example are: The object is a Bouguer Anomaly map; it is derived from gravity data; and the gravity data is from the southwest region of the U.S. These are all properties that can be verified against the map provenance. To explain how they are verified, we need to understand the difference between how IWSearch and general-purpose search engines work.

- *Looking for PML documents.* Like other search engines, IWSearch crawls the Web looking for documents to be indexes. However, IWSearch can differentiate Web pages that are PML documents from Web pages that are not PML documents.

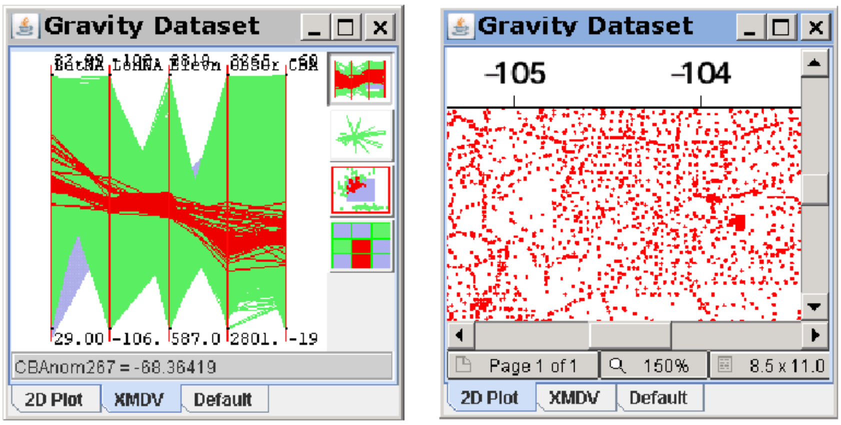
For the PML documents, IWSearch can get their content and use to populate a triple-store database that is capable of answering SPARQL queries [34]. This means that internally, IWSearch can consider the RDF triples inside PML documents and use SPARQL to query for specific properties about selected PML documents. For example, in a trivial way, we can use SPARQL to list responses for the `CreateContourMap` method in CM. In SPARQL terms, we would query for everything that is of type `NodeSet` and from these node sets we further query their `InferenceStep`s that uses the method `CreateContourMap`.

- *Select PML documents about Bouguer Anomaly maps.* A first SPARQL query would ask for PML documents that are node sets which have conclusions of the type `BouguerAnomalyMap`, e.g., the document in Figure 8). This would reduce drastically the number of PML documents that have the potential to answer the scientist's request.
- *Select Bouguer Anomaly maps derived from gravity data.* From this pool of Bouguer Anomaly maps, IWSearch performs a more complex and expensive combination of SPARQL queries that would use the relationships between node sets (the `hasAntecedents` property inside of inference steps) to traverse down the PML proof trace of each map and identify those maps that were derived from datasets of type `GravityData`. In this case, these gravity datasets are also the conclusions of other node sets reached in the process of traversing down the derivation paths of each Bouguer Anomaly map. This is a step that can be pre-processed for each PML node set added to the triple-store database.
- *Select gravity data from the southwest region of the U.S.* Finally, by inspecting the node sets holding the datasets of type `GravityData`, IWSearch can verify which ones are from the southwest region of the U.S. by verifying the parameter values used to retrieve the gravity data from the PACES database.

As one can see, semantic annotation enables the task of searching for scientific data and other products. Without the use of semantic annotations in the Bouguer Anomaly map example, it would be much harder if even possible for a scientist to locate the map of interest. This query capability addresses the **Challenge 4** in the use case. We claim that CI-Miner addresses Challenge 4 because the semantic annotations used to represent abstract workflows and provenance, i.e., WDOs, SAWs, and PML documents, are encoded in OWL and based on Web technologies that include URLs and namespaces. Further, CI-Miner uses novel approaches for searching and querying the content of these semantic annotations, e.g., triple-store technologies may be used to query WDOs, SAWs, and PML. In addition to the query capabilities mentioned above, it is worth mentioning that the queries considered in the CM use case are defined in terms of scientists' terminology: the `BouguerAnomalyMap` was originally defined in a workflow-driven ontology like the one presented in Figure 6 and developed by the scientists [22].

5.5 Visualization Capabilities

The capability of visualizing process components and results (**Challenge 5** in our use case), whether the results are intermediate or final, is as important as the capability of searching for these results. For example, for the gravity datasets provided by PACES in Figure 2, we can consider the use of three visualizations: textual view, plot view,



CBAnom267 = -68.36419

Fig. 9 Different viewers for gravity data sets.

and XMDV view. The default textual view is a table; the raw ASCII result from gravity database. The location plot viewer provides a 2D plot of the gravity reading in terms of latitude and longitude. XMDV, on the other hand, provides a parallel coordinates view, a technique pioneered in the 1970's, which has been applied to a diverse set of multidimensional problems [40]. Figure 9 shows a pop-up of the 2D plot and XMDV visualizations in their respective viewer windows. Upon selecting a node set in a gravity data derivation trace, provenance visualization tools like ProbeIt![8] [9] are able to determine, based on a semantic description of the output data, which viewers are appropriate. This is similar to a Web browser scenario in which transmitted data is tagged with a MIME-TYPE that is associated with a particular browser plug-in. These visualization tools should be flexible enough to support a wide array of scientific conclusion formats just as Web browsers can be configured to handle any kind of data, but also leverage any semantic descriptions of the data. For example, XMDV is a viewer suited to any n-dimensional data; the data rendered by XMDV need only be in a basic ASCII tabular format, as shown on the right hand side of Figure 9, with a few additional headers. Because gravity datasets are retrieved in an ASCII tabular format, XMDV can be used to visualize them. However, this kind of data is also semantically defined as being of type `GravityData`, in which case provenance visualization tools need to be configured to invoke the more appropriate 2D spatial viewer, as shown in the center of Figure 9. The semantic capabilities provided by these tools viewer need to compliment the MIME tables used in typical Web browsers, which only indicate the format or syntax of the data.

Semantic annotations captured by CI-Miner enable the task of visualizing scientific data and other products. Without the use of semantic annotations to identify the kind of data available in the gravity datasets in the example above, it would be difficult for scientists first to have access to these intermediate results and second to know which tool to use to visualize these results. This visualization capability corresponds to the **Challenge 5** in our use case. We claim that CI-Miner addresses Challenge 5 because of the following: the SAW specification includes a visual notation used to present abstract workflows to scientists; tools like Probe-It provide browsing capabilities for PML at the

[8] ProbeIt! tool available at http://trust.utep.edu/probeit/applet/

same time that it reuses conventional and state-of-the-art visualization capabilities to support the visualization and data analysis of scientific data and their provenance.

6 Conclusions and Future Work

This article introduces CI-Miner methodology for semantically enhancing scientific processes. The steps in the methodology take into consideration the need for scientists to be fully involved in semantic enhancements of scientific processes, whether processes are entirely performed by humans, executed by machines, or a combination. In CI-Miner, scientists can create domain-specific terminologies encoded as workflow-driven ontologies and can use these ontologies to support the specification of both abstract workflows about scientific processes and provenance about the outcome of scientific process executions. One of the goals of CI-Miner is to capture and preserve knowledge about scientific processes.

The use of both abstract workflows and provenance to annotate scientific data and products allow semantic-aware tools to support complex tasks, which has been identified as a challenge for conventional scientific processes. *Semantic data integration* allows scientists and semantic-enabled tools to figure out whether two datasets have the same type and, if not, whether two attributes in distinct are of the same type. Data integration processing leverages knowledge encoded in ontologies (including workflow-driven ontologies) and abstract workflows. *Semantic search* leverages the knowledge encode in provenance allowing scientists to look for specific components of scientific processes. *Semantic visualization* allows scientists to visually analyze and understand many components of scientific processes, including diagrams representing the process specifications. The approach supports multiple visualization strategies for each process result. The methodology has the potential of enabling scientific communities to take advantage of an entire new generation of semantically-enabled tools and services that are under development by industry and academic organizations.

CI-Miner was originally designed to enhance scientific processes to support interdisciplinary projects at UTEP's CyberShARE Center[9] that required dealing with multiple scientific processes that are at a different level of maturity, applied to a distinct scientific field, i.e., geoscience, environmental sciences and computational mathematics, and supported by a distinct community. CI-Miner has been a collective effort among the communities directly involved with CyberShARE and with members of the Mauna Loa Solar Observatory at the National Center of Atmospheric Research. One of the most important results of CI-Miner to date is that its development has brought distinct communities closer, making possible for them to collaborate and share resources.

Acknowledgements This work was supported in part by NSF grants HRD-0734825 and EAR-0225670.

References

1. Aldouri, R., Keller, G., Gates, A., Rasillo, J., Salayandia, L., Kreinovich, V., Seeley, J., Taylor, P., Holloway, S.: GEON: Geophysical data add the 3rd dimension in geospatial studies. In: Proceedings of the ESRI International User Conference 2004, p. 1898. San Diego, CA (2004)

[9] http://cybershare.utep.edu/

2. Ashish, N., Knoblock, C.: Wrapper generation for semi-structured Internet sources. ACM SIGMOD Record **26**(4), 8–15 (1997)
3. Baker, P.G., Goble, C.A., Bechhofer, S., Paton, N.W., Stevens, R., Brass, A.: An ontology for bioinformatics applications. Bioinformatics **15**(6), 510–520 (1999). DOI 10.1093/bioinformatics/15.6.510. URL http://dx.doi.org/10.1093/bioinformatics/15.6.510
4. Barga, R.S., Digiampietri, L.A.: Automatic capture and efficient storage of e-science experiment provenance. Concurrency and Computation: Practice and Experience **20**(5), 419–429 (2008)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
6. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Franc o.Y.: Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation (2008)
7. Clifford, B., Foster, I.T., Vöckler, J.S., Wilde, M., Zhao, Y.: Tracking provenance in a virtual data grid. Concurrency and Computation: Practice and Experience **20**(5), 565–575 (2008)
8. Davidson, S.B., Cohen-Boulakia, S., Eyal, A., Ludascher, B., McPhillips, T.M., Bowers, S., Anand, M.K., Freire, J.: Provenance in scientific workflow systems. IEEE Data Eng. Bull. **30**(4), 44–50 (2007)
9. Del Rio, N., Pinheiro da Silva, P.: Probe-it! visualization support for provenance. In: G. Bebis, R.D. Boyle, B. Parvin, D. Koracin, N. Paragios, T.F. Syeda-Mahmood, T. Ju, Z. Liu, S. Coquillart, C. Cruz-Neira, T. Müller, T. Malzbender (eds.) ISVC (2), *Lecture Notes in Computer Science*, vol. 4842, pp. 732–741. Springer (2007)
10. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, pp. 652–659. ACM, New York, NY, USA (2004). DOI http://doi.acm.org/10.1145/1031171.1031289
11. EarthScope. http://www.seis.sc.edu/ears/
12. Edwards, P.N., Jackson, S.J., Bowker, G.C., Knobel, C.P.: Understanding infrastructure: Dynamics, tensions, and design. Tech. rep., School of Information, University of Michigan (2007). Final report of the workshop, "History and Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures"
13. Fonseca, F.T., Davis, C.A., Câmara, G.: Bridging ontologies and conceptual schemas in geographic information integration. GeoInformatica **7**(4), 355–378 (2003)
14. Gates, A.Q., Pinheiro da Silva, P., Salayandia, L., Ochoa, O., Gandara, A., Del Rio, N.: Use of abstraction to support geoscientistsúnderstanding and production of scientific artifacts. In: G. Keller, C. Baru (eds.) Geoinformatics: Cyberinfrastructure for the Solid Earth Sciences. Cambridge University Press (To appear)
15. The Geosciences Network: Building Cyberinfrastructure for the Geosciences. http://www.geongrid.org/
16. Gil, Y., Ratnakar, V., Deelman, E., Spraragen, M., Kim, J.: Wings for pegasus: A semantic approach to creating very large scientific workflows. Proceedings of the OWL: Experiences and Directions(OWLED-06), Athens, GA, November 10-11, (2006)
17. Giunchiglia, F., Shvaiko, P.: Semantic matching. Tech. Rep. Vol. 71, CEUR - WS (2003). URL http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-71/Giunchiglia.pdf
18. The gene ontology. http://www.geneontology.org/
19. Guarino, N.: Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In: Proceedings of SCIE, pp. 139–170 (1997)
20. Jösang., A., Knapskog, S.: A metric for trusted systems. In: Proceedings of the 21st NIST-NCSC National Information Systems Security Conference, pp. 16–29 (1998)
21. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. Proceedings of the 12th international conference on World Wide Web pp. 640–651 (2003)
22. Keller, G., Hildenbrand, T., Kucks, R., Webring, M., Briesacher, A., Rujawitz, K., Torres, R., Gates, A., Kreinovich, V.: A Community Effort to Construct a Gravity Database for the United States and an Associated Web Portal. In: Special Paper on Geoinformatics. Geological Society of America (2004)
23. Kim, J., Deelman, E., Gil, Y., Mehta, G., Ratnakar, V.: Provenance trails in the wings/pegasus system. Concurrency and Computation: Practice and Experience **20**(5), 587–597 (2008)

24. Kushmerick, N.: Wrapper Induction for Information Extraction. Ph.D. thesis, University of Washington (1997)
25. McGuinness, D., Ding, L., Pinheiro da Silva, P., Chang, C.: PML2: A Modular Explanation Interlingua. In: Proceedings of the AAAI 2007 Workshop on Explanation-aware Computing. Vancouver, British Columbia, Canada (2007)
26. McGuinness, D.L.: Ontological Issues for Knowledge-Enhanced Search. In: N. Guarino (ed.) Proc. of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98), pp. 302–316. IOS Press, Trento, Italy (1998)
27. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. Tech. rep., World Wide Web Consortium (W3C) (2004). URL http://www.w3.org/TR/owl-features/. Recommendation
28. McGuinness, D.L., Pinheiro da Silva, P.: Explaining Answers from the Semantic Web. Journal of Web Semantics **1**(4), 397–413 (2004)
29. Miles, S., Groth, P., Munroe, S., Moreau, L.: Prime: A methodology for developing provenance-aware applications. ACM Transactions on Software Engineering and Methodology (2009). URL http://eprints.ecs.soton.ac.uk/17450/
30. NASA/Science Office of Standards and Technology: Definition of the Flexible Image Transport System (FITS), Standard NOST 100-2.0 (1999)
31. Pinheiro da Silva, P., McGuinness, D.L., Del Rio, N., Ding, L.: Inference web in action: Lightweight use of the proof markup language. In: International Semantic Web Conference (2008)
32. Pinheiro da Silva, P., Salayandia, L., Gates, A.: Wdo-it! a tool for building scientific workflows from ontologies. Tech. Rep. UTEP-CS-07-50, University of Texas at El Paso, El Paso, TX (2007)
33. Pinheiro da Silva, P., Sutcliffe, G., Chang, C., Ding, L., Del Rio, N., McGuinness, D.: Presenting TSTP Proofs with Inference Web Tools. In: R. Schmidt, B. Konev, S. Schulz (eds.) Proceedings of the Workshop on Practical Aspects of Automated Reasoning, 4th International Joint Conference on Automated Reasoning, p. Accepted. Sydney, Australia (2008)
34. Prud'hommeaux, E., Seaborne, A.: Sparql query language for rdf. W3C Recommendation (2008)
35. Salayandia, L., Pinheiro da Silva, P., Gates, A.Q., Salcedo, F.: Workflow-Driven Ontologies: An Earth Sciences Case Study. In: Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing. Amsterdam, Netherlands (2006)
36. Simmhan, Y.L., Plale, B., Gannon, D.: Karma2: Provenance management for data-driven workflows. Int. J. Web Service Res. **5**(2), 1–22 (2008)
37. Semantic web for earth and environmental terminology (sweet). http://sweet.jpl.nasa.gov
38. Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M.: Workflows for e-Science: Scientific Workflows for Grids. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
39. Wroe, C., Stevens, R., Goble, C., Roberts, A., Greenwood, M.: A suite of daml+oil ontologies to describe bioinformatics web services and data. International Journal of Cooperative Information Systems special issue on Bioinformatics **12**(2), 597–624 (2003)
40. Xie, Z.: Towards Exploratory Visualization of Multivariate Streaming Data. http://davis.wpi.edu/
41. Zaihrayeu, I., Pinheiro da Silva, P., McGuinness, D.L.: IWTrust: Improving User Trust in Answers from the Web. In: Proceedings of 3rd International Conference on Trust Management (iTrust2005), pp. 384–392. Spring, Paris, France (2005)
42. Zhao, J., Goble, C.A., Stevens, R., Turi, D.: Mining taverna's semantic web of provenance. Concurrency and Computation: Practice and Experience **20**(5), 463–472 (2008)